AD A024824

9

SEMI-ANNUAL REP[redacted]T.

1 Jul - 31 Dec 75.

(12)

sponsored by

ADVANCED RESEARCH PROJECTS AGENCY

ARPA Order No. 2245 | Program Code No. P3D20

15

U.S. ARMY CONTRACT DAHC-15-73-C-0077, ARPA Order-2245

Effective: August 1, 1972 thru September 30, 1976

6

DEMONSTRATION AND EVALUATION OF THE PLATO IV COMPUTER-BASED EDUCATION SYSTEM.

(Computer-based Education for a Volunteer Armed Service Personnel Program).

(12) 179p.

DDC

MAY 26 1976

B

For the Period

July 1, 1975 -- December 31, 1975

AD017399

Computer-based Education Research Laboratory

University of Illinois Urbana Illinois

408 130

11

Mar[redacted]976

PREFACE

This report under Contract DAHC 15-73-C-0077 describes a program aimed at the demonstration, test and evaluation of the educational and economic effectiveness of the PLATO ~~IV~~ 4 computer-based education as implemented in several geographically dispersed military training sites. It also describes a program aimed at increasing the cost effectiveness of the PLATO system, both in its deployment in the ARPA community and in its continuing development as a national resource for education.

TABLE OF CONTENTS

I. SITE SUPPORT PROGRAM FOR SERVICE TEST OF PLATO IV

II. TECHNICAL PROGRAM

PART I

SITE SUPPORT PROGRAM FOR SERVICE TEST OF PLATO IV

INTRODUCTION TO SITE SUPPORT PROGRAM

December 31, 1975, marked the end of the third full year of the ARPA/PLATO project. During 1975, authors at the ARPA/PLATO sites completed approximately 115 lessons on the PLATO IV system. About half of these lessons were used in researching various aspects of computer-based education. The other half were written at Chanute Air Force Base (for training Special Purpose Vehicle Repairmen) and at Sheppard Air Force Base (to train Physician Assistants). During this year, 189 students have successfully completed Chanute's lesson sequence while logging 5,670 hours on the PLATO system. At Sheppard, where the students are still in the midst of a one-year course, the lessons have been used by 32 students for over 36 hours. At Aberdeen, 178 students logged over 2,000 hours as that site completed its evaluation.

Since the ARPA sites had moved into an evaluation phase, the efforts of the Military Training Center (MTC) support group were augmented by those of the PLATO Educational Evaluation and Research (PEER) group. The following report summarizes the highlights and major events of the second half of 1975.

1.1 MTC ACTIVITIES

1.1.1 TRAINING

The MTC "Introduction to PLATO" course was used to train individuals from Chanute Air Force Base and Fort Monmouth. The Chanute training, lasting about three weeks, encompassed active participation in lesson review and revision.

Over the entire year, MTC trained 16 ARPA/PLATO authors. Twenty additional authors were trained by site personnel using MTC training materials.

1.1.2 CONSULTATION

In addition to the usual amount of on-line TUTOR and instructional design consultation, CERL staff have spent increasing amounts of time consulting on statistical and evaluation questions, primarily with Chanute Air Force Base. Such consultation, together with programming assistance, has resulted in an average of 1-2 meetings per week for the last three months of the year.

1.1.3 SUPPORT PROGRAMMING

Prior to this reporting period, the phrase "support programming" was defined as TUTOR programming performed by CERL staff for remote ARPA/PLATO sites. With the growing importance of evaluation activities at Chanute, MTC and PEER personnel began offering assistance in writing programs in languages other than TUTOR. For example, the SOUPAC programs employed to analyze Chanute's data were written by an MTC group member. In addition,

Chanute's data had been formatted incorrectly for SOUPAC and conversion programs (written in PL/1 and FORTRAN) were needed.

While carrying out this new type of support programming, MTC continued to offer assistance in the use of TUTOR and to modify and create programs for the sites. The major TUTOR programming efforts were:

Test Driver

The test driver described in previous annual reports has been successfully incorporated into many Sheppard lessons. The driver allows for 20 items per test which is sufficient for lesson criterion tests. However, Sheppard authors also needed an end-of-trimester test package that could handle up to 50 multiple choice or short answer test questions. To meet this need, MTC programmed and tested an expanded form of the test driver with an increased number of test items and a data analysis package which provides both graphical and numerical information on individual student and item performance.

ECS Monitor

To enable Chanute personnel to schedule their terminal use to coincide with memory space availability (ECS), MTC designed and implemented a program to continuously monitor ECS usage and graph the result. The program is designed to be automatically read into memory and is executed whenever the

computer is operating. In this way, data loss through system crashes or human factors is minimized.

Technical Orders

In October, Chanute personnel expressed interest in using PLATO to simulate a Technical Order (TO) library. Such a simulation would give the student a chance to practice finding TO's while the computer tracks his progress, diagnoses his weaknesses, and prescribes appropriate study or remediation. The simulated TO library requires a large database of genuine and "created" TO's which can be referenced by various drills and teaching sequences. To allow Chanute to quickly amass the database, CERL staff wrote an editor to store and recall TO displays. Clerical staff at Chanute will be able to input quickly the information needed to create the TO database. If the TO lessons are used in other courses, additional databases can be generated easily.

1.1.4 LESSON REVIEW

The MTC book Lesson Review has been published and a copy has been distributed to all PLATO sites.

Major reviewing jobs for Sheppard and Chanute have occupied MTC reviewers during this reporting period. The 57 reviews of Sheppard lessons are described in the Sheppard Air Force Base site report (section 1.1.2).

As part of the Chanute evaluation, MTC agreed to review nine lessons which were selected by Chanute to fall into three categories. Three of the lessons met the validation criterion (27 of 30 students with 75-80% of posttest items correct) essentially the first time they were used on full classes of students. Three more lessons met this criterion after some modification to the original lessons. At the inception of the study, three lessons had not met the validation criterion. MTC obtained copies of the nine lessons before and after any modifications had been made. MTC is now in a process of reviewing and analyzing the differences between the three groups of lessons as well as between "before" and "after" versions of the lessons. The reviewer performing the analyses was not told which lessons were easily validated or which required changes. Furthermore, he analyzed the "before" versions of the lessons prior to examining the "after" versions. This study is intended to cast some light on the validity of MTC reviews as well as to describe more completely certain aspects of Chanute lesson development practices.

MTC personnel reviewed several of Maxwell Air Force Base's lessons. See section 1.2.4 for a description of these activities.

1.1.5 TASK PLANNING

Throughout this reporting period, about one man-year was devoted to an extensive analysis and revision of the goals of CERL support. The short-term product was a series of six concept papers which were used as a basis for interaction with sites and sponsor.

Each concept paper served as the focus for meetings between CERL and sponsor and/or sites, the results of which were used in an iterative

fashion to produce the next concept paper. In late December, the process yielded specifications for a research program sensitive to the needs of both sponsor and sites.

1.1.6 LIAISON

The rewriting of an existing MTC lesson and the broadening of MTC's file management capabilities have enabled MTC to improve its service in furnishing disk space to the ARPA/PLATO sites. Since July 1974, these sites have requested additions or deletions of disk space, name changes, creation of special types of lessons, etc. through lesson "mtcrequest." In September 1975, a user from the Naval Training Equipment Center at Orlando, Florida, rewrote this lesson. In its new form, "mtcrequest" provides a medium through which the users at remote sites can easily make requests for disk space and by which they can be informed of the availability of their lessons once the requested files have been created by MTC. For the MTC group, "mtcrequest" serves as an automated log of its file management activities. Soon after the new version of "mtcrequest" was put into use, new system level programs gave MTC the capability to carry out the operations with disk space that were requested in "mtcrequest." Previously, these operations were done by the systems staff so that immediate response to site needs was frequently impossible. Together these changes have improved the smoothness and reliability of file space handling.

The ARPA/PLATO sites frequently give demonstrations of the PLATO IV system to visiting personnel. To assist the sites in this activity, MTC provides the names of lessons that demonstrate PLATO capabilities, insures that the sites have enough ECS to carry out the demonstration, supplies the

sites with microfiche and handouts, and provides active assistance in the demonstration itself either through the communications capabilities of the system or through a site visit.

The PLATO system continues to be the primary medium through which the sites request and MTC provides liaison and other services. The fact that roughly 70 ARPA/PLATO authors sent over 7000 notes on the system in the last half of 1975 is indicative of this fact.

1.2 INTERACTIONS WITH THE SITES

1.2.1 NEW SITES

New sites at Redstone Arsenal, Fort Eustis, and the Air Force Academy were established since mid-year. Although MTC has not provided formal TUTOR training to these new sites, training manuals and materials were supplied, and authors at the sites were given an on-line orientation to acquaint them with MTC and CERL procedures.

MTC hosted 3 staff members from Wright-Patterson who made a 2-day visit to CERL. The purpose of the visit was to prepare for the acquisition of a terminal (on loan from Maxwell Air Force Base) in February. In anticipation of obtaining a terminal, one of the visitors had received TUTOR training from MTC while stationed at Maxwell.

1.2.2 SCHOOL OF HEALTH CARE SCIENCES SHEPPARD AIR FORCE BASE

On 2 July 1975, the first group of 16 first trimester Physician Assistant students at the School for Health Care Sciences began to use the materials developed by the PLATO authors at the School. During the remainder of the year, some of this group completed first trimester lessons and began the second trimester. In November, a second wave of 16 students began the first trimester lessons. During the span of this report, the Sheppard authors tested students' end-of-lesson performance, administered instruction and testing to the PLATO Physician Assistant students, refined lessons where student performance indicated a need, and developed new lessons for the second and third trimesters. Shpeast, MTC's support group for Sheppard, worked to supplement those efforts.

The largest component of support consisted of lesson reviews which assisted authors in revising lessons to incorporate sound educational strategies and techniques as well as to avoid actual and potential difficulties for students. Since 1 July 1975, Shpeast has completed 57 lesson reviews for Sheppard. Seven of these reviews were done with the author monitoring the reviewer and communicating via the "talk" option while the reviewer went through the lesson as a student. The remainder of the reviews were written critiques, enumerating both general and specific considerations for lesson improvement.

Though most of Sheppard's reviews have been delivered on-line, site visits either by MTC staff to Sheppard AFB or by Sheppard authors to CERL have provided an occasion for extensive, tête-á-tête reviews. These reviews were the major consulting activity Shpeast carried out in the 16 man-days Shpeast personnel visited Sheppard AFB and the 5 man-days Sheppard personnel visited CERL.

Throughout this half, Shpeast also provided programming support in day-to-day coding problems, use of system features, and the creation or reprogramming of test administration and analysis packages (see section 1.1.3, Support Programming).

1.2.3 LEARNING RESOURCES CENTER FORT BELVOIR

During this reporting period, personnel from the Army Research Institute used the four terminals at Fort Belvoir to conduct research in educational psychology. MTC support of this research consisted of assistance with student routing and data collection techniques. In September, an MTC group member visited Fort Belvoir and met with personnel of the Learning Resources Center. As a result, the site director was provided with a

list of lessons available on the system for supplementing courses taught at Fort Belvoir. With the MTC training materials and on-line consultation, 3 new authors have been trained at Fort Belvoir.

1.2.4 AIR UNIVERSITY MAXWELL AIR FORCE BASE

At the end of the last reporting period, MTC personnel had spent two weeks training Maxwell authors on-site. In August of 1975, MTC personnel returned to Maxwell to conduct an advanced TUTOR training course, an instructional design course including seminars, and on-line and written lesson reviews. MTC personnel also described and demonstrated microfiche preparation and suggested several procedures and checklists to streamline lesson development.

1.2.5 CHANUTE TECHNICAL TRAINING CENTER CHANUTE AIR FORCE BASE

Following the February 1975 program review and the subsequent acceptance of a research program for the continuation period, Chanute staff selected two of the proposed projects as especially important: Technical Order training and simulation of the Sun Corporation's Engine Analyzer. In addition, at the midyear point, about 20% of the lessons written previously had not met the validation criterion. Chanute's efforts this half year have been divided between modifying previously written lessons for validation and writing new lessons. The last lessons were validated in December. Of the new materials, the TO sequence is the more completely developed with several lessons programmed. No lessons for the Engine Analyzer are beyond the planning stage.

Liaison

Parkland Community College in Champaign has begun to use a portion of Chanute's vehicle training curriculum. MTC coordinated the use of appropriate routers and arranged for use of microfiche at Parkland.

MTC has also begun an investigation of the problems relating to the use of microfiche at Chanute. The problem of lost visual detail has been diagnosed: it is caused by the high contrast microfiche film used by CERL coupled with strong reflections obtained on the 35mm images when artificial light is used to illuminate the polished metal parts found in an automobile. The contrast of the 35mm images can be reduced rather easily by use of a special highlight masking film. A quantity of this film has been obtained by CERL and delivered to Chanute personnel.

Evaluation Support

The last student data for inclusion in the Chanute evaluation was collected on September 30, 1975. A team of Chanute and CERL evaluators has completed the bulk of the analysis of the student data, and efforts are now directed at interpreting and reporting the results.

CERL has provided Chanute with support in analysis of data in both the Instructional Impact and Instructional Effectiveness areas of their evaluation. This support has included punching and verifying of data cards, identification of cases with incomplete data, preparation and debugging of

SOUPAC[1] program control decks, and assistance in interpretation of SOUPAC program output. Thus, CERL has provided both personnel and computing system resources in support of the Chanute evaluation.

The Instructional Impact data which was available for analysis during the reporting period consists of student and instructor responses to attitude questionnaires. A 6-item attitude survey was administered to students in both PLATO and non-PLATO conditions at the end of each of the four blocks of instruction in the common course segment of the Special Purpose Vehicle Repairman course. In addition, a more comprehensive 66-item attitude survey was administered to students at the end of the common course segment. Instructor attitudes were sampled twice -- once rather early in the implementation period and a second time after the computer-based training system had been in use for several months.

A variety of statistical analyses were conducted on the attitude data including both descriptive and inferential techniques. Means, standard deviations and frequencies for each response category were computed on all variables for course instructors and for students in each experimental condition. The responses to the 66-item student attitude questionnaire were subjected to a principal axis factor analysis followed by

---

[1]SOUPAC (Statistically Oriented Users Programming and Consulting) is a system of statistical analysis routines developed and supported by the University of Illinois Computing Services Office.

varimax rotation. Iterative factor analysis and oblimax rotation were also applied in an attempt to obtain interpretable factors. Since the number of instructors who completed the instructor attitude survey was less than the number of items it contained, factor analysis of instructor attitudes was not possible. In addition to the analyses already described, Thurstone's method of unidimensional scaling was applied to student responses regarding their attitudes toward PLATO as compared to other media (e.g., programmed texts, workbooks, film, classroom lectures, laboratory activities).

A variety of measures of instructional effectiveness were made by Chanute both prior to and during the implementation of the computer-based training system. The scores made by students on exams following each block of instruction in both the common and individual shred segments of the target courses were recorded for a baseline (BL) group who had completed training before PLATO and Instructional System Design (ISD) changes to the courses were introduced. Similar block exam scores were recorded for students who were trained during the implementation period under either the PLATO-based (PB), conventional/PLATO (CP), or non-PLATO (NP) training methods. Performance on block exams is thus one of the major criteria by which the instructional effectiveness of the computer-based training system can be judged.

In addition to the block exams, Chanute recorded performance of students in the PB, CP, and NP conditions on a

Special Topics Test which was administered three times -- once as a pretest before instruction, once as a posttest following the portion of the courses which is taught in common, and once at the end of the specialty shreds. This too is a direct measure of instructional effectiveness. Measures which are less direct include absenteeism, rate of washbacks, rate of eliminations, amount of remedial instruction, and time required to complete each block.

As with the attitude measures, both descriptive and inferential statistical techniques were applied to the available data on instructional effectiveness. Repeated measures analysis of variance was used to assess the effects of method of training on block exam and Special Topics Test performance. Multivariate analysis of covariance has also been applied to the Special Topics Test data and the results of this analysis will be compared with those for the repeated measures analysis. Chi square analyses of the frequency of washbacks, elimination, and absenteeism have also been done where feasible. The performance of students on individual lesson exams, the Master Validation Exams, or MVE's was recorded by Chanute and analyzed by CERL. In particular, the frequency of scores above and below the validation criterion was obtained for students completing lessons which had previously been validated (i.e., those for which 90% of a sample of 20-30 consecutive students had performed at or above the standard implied by the lesson's objective). Estimates of the reliability of the Special Topics Test were also computed.

1.3

SUMMARY INFORMATION

For a detailed explanation of the following tables, see section 3.2 "Summary Information" in the Second Annual Report.

System usage is monitored by a sampling procedure which surveys every terminal once each hour that the system is in operation (including holidays, weekends, and test periods). This information is provided for use by the PEER group.

1.3.1
Site Data

Site	Total # Authors	Authors > 3 hrs/wk.	Author Hours 7/1 - 12/30	Disk Space Used	Terminals 12/30	Visits to Sites		Visits to CERL	
						Trips	Man-Days	Trips	Man-Days
Aberdeen[1]	5	2	77	48	0				
AF Academy	6	3	463	5	2				
ARI	8	2	1171	98	2	1	1		
ARPA	1	0	n.a.	0	1	2	7		
Belvoir	5	0	1491	16	4	1	1		
Chanute	11	5	1133	243	30	1[2]	3	2[2]	4
Eustis[3]	1	1	89	31	3				
HumRRO	10	3	684	44	2				
ISI-USC[1]	4	1	850	2	0				
Lowry	10	5	1133	34	4	2	11		

[1]This site disbanded this 6 month period.

[2]In addition to the formal meetings noted here, Chanute and CERL staff have met 1-2 times/wk. for consultation on programming and evaluation.

[3]Data for Ft. Eustis includes usage totals for staff members who followed the terminals from Ft. Monmouth.

1.3.1
Site Data

Site	Total # Authors	Authors > 3 hrs/wk.	Author Hours 7/1 - 12/31	Disk Space Used	Terminals 12/31	Visits to Sites		Visits to CERL	
						Trips	Man-Days	Trips	Man-Days
Maxwell	6	4	1226	90	4	1	7	1	6
MTC	14	14	1956	136	3				
Monmouth	1	1	101	31	1			1	4
NPRDC	16	7	1630	111	12				
Orlando	9	5	1043	43	4				
Redstone	7	3	1718	6	4				
Sheppard	12	12	3349	278	20	3	16	2	5
USC	6	1	260	39	2				
UCSB	1	1	150	0	1				
TOTAL	133	70	18524	1255	101[1]	11	46	6	19

[1] This figure does not include one terminal at University of Illinois Psychology Department, two terminals in the Electrical Engineering Department, or two terminals at the Educational Testing Service (ETS).

1.3.2
Terminal Delivery and Re-Assignment
6/75 - 12/75

Site	6/75	7	8	9	10	11	12	Total
abe	14	-7	-3	-4				0
afa	0			2				2
ari	1		3	-2				2
arpa	1							1
bel	4							4
cha	23	7						30
eus	0			3				3
hum	2							2
isi	1							1
low	4							4

Site	6/75	7	8	9	10	11	12	Total
max	4							4
mtc	3							3
mon	4			-3				1
NPRDC	12							12
orl	4							4
red	0			4				4
shp	20							20
usc	3							3
ucsb	1							1

1.3.3.1

PLATO IV - Terminal Usage

July 1, 1975 - September 30, 1975

User	Terminals September 1975	Mean Hours per Week per Terminal July Prime	July Total	August Prime	August Total	September Prime	September Total
CERL	67	24.2	33.2	23.8	34.1	25.0	35.4
U of I	291	16.4	22.5	16.0	23.1	34.2	48.6
Ill. Univ.'s	78	18.2	22.1	13.7	17.5	15.8	18.7
Other Univ.'s	80	29.5	43.9	25.8	40.1	37.0	57.4
Comm. Coll.'s	118	13.6	13.5	8.7	9.1	16.5	16.3
Schools	102	2.0	2.0	0.7	0.8	2.8	2.8
Government	31	17.3	20.5	16.9	21.8	15.9	20.6
Military	104	16.2	17.3	18.6	21.0	17.0	18.7
Commercial	12	29.6	37.6	30.5	40.3	24.2	31.8

October 1, 1975 - December 31, 1975

User	Terminals December 1975	Mean Hours per Week per Terminal October Prime	October Total	November Prime	November Total	December Prime	December Total
CERL	72	25.5	35.8	24.0	34.4	25.0	36.9
U of I	260	34.4	49.8	26.0	41.5	26.5	42.4
Ill. Univ.'s	136	22.0	28.7	20.6	28.5	19.1	27.8
Other Univ.'s	67	38.9	58.9	35.3	56.0	32.2	54.1
Comm. Coll.'s	118	28.8	28.9	23.4	23.5	25.6	25.8
Schools	103	6.4	6.5	10.2	10.2	12.1	12.1
Government	24	20.2	25.9	21.6	27.3	20.9	24.7
Military	98	16.8	18.0	17.1	19.2	19.7	22.9
Commercial	15	24.3	38.9	20.1	29.9	21.5	31.5

1.3.3.2

ARPA Terminal Usage

July 1, 1975 - September 30, 1975

User	Terminals September 1975	Mean Hours per Week per Terminal July Prime	July Total	August Prime	August Total	September Prime	September Total
ARPA-MAIN	2	6.8	6.7	12.0	12.6	6.6	6.5
ARI	1	36.1	37.9	34.8	37.8	30.4	31.3
ARPA-CERL	2	25.4	27.1	19.1	25.6	35.6	47.9
Air Force Academy	2	---	---	---	---	5.6	6.4
Aberdeen	-	6.7	6.7	---	---	---	---
Chanute	30	20.2	20.1	21.5	22.4	19.2	19.3
Monmouth	2	6.8	7.2	3.6	4.0	0.6	0.6
HumRRO	2	6.0	8.8	5.6	9.9	12.8	21.8
Lowry	3	19.8	25.7	19.9	23.0	25.1	25.2
San Diego	12	13.3	14.3	13.5	14.6	15.1	16.6
Sheppard	20	13.7	14.6	17.2	18.8	18.3	18.8
USC	8	7.2	7.6	12.9	17.3	7.5	11.4
Maxwell	4	28.4	30.5	28.0	33.6	23.6	26.3
Orlando	4	27.8	31.6	11.8	12.9	21.4	24.8
Belvoir	4	48.6	54.9	52.8	62.1	32.2	39.6
Eustis	4	---	---	---	---	2.8	4.2
Redstone	4	---	---	---	---	9.6	10.5

October 1, 1975 - December 31, 1975

User	Terminals December 1975	Mean Hours per Week per Terminal October Prime	October Total	November Prime	November Total	December Prime	December Total
ARPA-MAIN	1	5.1	5.1	9.6	9.6	15.8	15.8
ARI	1	32.8	33.9	31.6	31.9	33.6	34.0
ARPA-CERL	2	31.1	33.0	27.6	30.5	33.2	39.4
Air Force Academy	2	14.1	19.3	22.8	31.2	22.0	30.3
Aberdeen	-	---	---	---	---	---	---
Chanute	30	17.6	17.9	14.9	15.6	18.6	20.2
Monmouth	1	16.7	18.2	23.0	24.7	27.1	48.3
HumRRO	2	11.6	20.2	14.1	24.9	21.4	36.1
Lowry	3	31.0	31.4	23.5	25.9	20.0	20.2
San Diego	12	15.2	17.7	16.7	18.9	18.2	23.0
Sheppard	20	15.2	15.9	15.9	17.0	17.9	20.1
USC	4	6.1	6.1	4.2	4.4	15.1	15.4
Maxwell	4	22.2	24.3	22.3	26.1	24.7	27.8
Orlando	4	9.0	9.4	17.2	19.6	22.4	23.5
Belvoir	4	30.0	33.0	26.4	32.4	25.2	35.0
Eustis	4	2.7	4.2	5.0	6.3	6.6	8.5
Redstone	4	25.5	26.6	36.6	45.1	30.8	36.5

1.3.3.3

PLATO IV System Performance*

(July 1, 1975 - December 31, 1975)

	July	Aug.	Sep.	Oct.	Nov.	Dec.
Mean hours to interruption	9.59	13.24	17.31	8.16	18.02	11.26
**Mean down time (hours)	.34	.13	.14	.27	.21	.29
Proportion of class hours interrupted once or more	.12	.07	.06	.12	.06	.09
Weeks to terminal failure	13.90	7.00	3.70	6.30	9.20	13.40
Days to repair a terminal	2.40	1.60	2.10	1.70	1.30	.90
% of terminal-hours usable	93.20	95.80	91.20	93.00	96.70	96.90
Terminal hours used	76198	64941	112974	121438	104683	74736

*Times are prime times for the indicated months. In general 70 to 80 hours were scheduled for such non-experimental use each week.

**Down time includes time required for 99% of users to return to normal operation.

ARPA/PLATO PROJECT

Daniel Alpert - Director

The MTC Group

Larry Francis - Coordinator

Ken Barr

Don Emerick

Kathy Geissler

Alec Himwich

Jim Krakower

Joe Klecka

Lynn Misselt

Susy Soung

Eileen Sweeney

Mark Swenson

The PEER Group

Allen Avner - Chief

Esther Steinberg

Kumi Tatsuoka

PART II

TECHNICAL PROGRAM

INTRODUCTION TO THE TECHNICAL PROGRAM

The technical program at CERL has for over a decade been guided by considerations of both performance and cost in the delivery of high quality education through the interactive use of computers. This work has led to a new display device, the Plasma Display Panel, a new interactive graphics-oriented language, TUTOR, and a new architecture for information processing. What is perhaps most important is that these and other developments fit together in a highly effective system which is greater than the sum of its parts. Part II of this ARPA report describes the status of a program that with ARPA support is maintaining the momentum of technical development at CERL.

2. AUXILIARY MASS STORAGE

The Auxiliary Mass Storage (AMS) project involving semiconductor serial shift registers has been the testing and proving ground for memory architecture ideas as well as a valuable test bed for the development of high speed memory and memory controller techniques. Serial shift registers are not, however, the most cost effective memory devices available today due to recent advances within the semiconductor industry. Large bit per chip Random Access Memories (RAMs) have been developed to the point of good availability and high manufacturing yields. Semiconductor systems (including cabinetry, power supplies, etc., but not including memory controllers for the Control Data Extended Core Storage [ECS] architecture) are available for 0.3¢ per bit. This is the same approximate price for the semiconductor devices themselves in the early AMS system.

Presently, studies are being made to determine the best system placement of a new RAM implemented AMS. Two diverse directions are being studied.

The first possible position of an AMS II (a RAM implemented AMS) would be in the same hierarchical position as the AMS I, as the source of high speed data into and out of an ECS port of which there are four. In this fashion, the AMS memory system would be second order removed from the basic PLATO system and therefore more simply worked on without interference from an operating PLATO system. However, just as in this position it would be second order removed in a hardware sense, so too it would be second order removed in a software sense, requiring

double data transfers (AMS to ECS followed by ECS to Central Memory [CM]) for Central Processing Unit (CPU) access to AMS stored data.

The second possible position of an AMS II would be as an expandable replacement for the actual ECS bay. The CDC ECS system operates as a multitude of "bays" (maximum of 4), each of which contains 1/2 million ECS words. These ECS bays communicate directly (via the ECS controller) to CM upon the execution of an ECS read or ECS write instruction in the CPU(s). The second proposed hierarchical position of the AMS II memory block would be in place of one of the 1/2 million word ECS "bays". The AMS block would not, however, be limited to the 1/2 million word size as it could be designed to comprise any number of 1/2 million word "pages" (books?) and the exact page number presented to the ECS controller could be programmable. Most likely word zero of all pages would indicate the page being accessed and the CPU could select a page by writing a page number to this location independently of the present page. The positioning of AMS memory at this level in the PLATO memory structure has the one significant advantage that data held in AMS could be directly accessed by the CPU via ECS read and ECS write instructions. The major disadvantage of this placement is that the first 1/2 million words of AMS II would cause exactly zero increase in the total memory space available. Instead, it would simply replace the existing ECS bay.

The two approaches to the AMS II placement are being studied from both the hardware and software viewpoints.

P. Tucker
L. Hedges
D. Anderson

3. A LOW COST, HIGH EFFICIENCY POWER SUPPLY FOR PLASMA DISPLAY TERMINALS

In present Plasma Display Terminals, the maximum utilization of the Plasma Display cannot be achieved partially because the use of conventional power supplies for both the actual display and the support circuitry limits the minimum size of the terminal. A new, high efficiency power supply is being developed to overcome this problem.

The plasma terminal power supply has the following constraints:

Input Voltages: 100-130V AC 60/50 Hz

Output Voltages:

130V:	sustainer supply @ .6A max.	(78W)
18V:	pulse/driver supply @2.0A max.	(36W)
12V:	logic supply @1.0A max.	(12W)
-5V:	logic supply @ .2A max.	(1W)
5V:	logic supply @5.0A max.	(15W)
10V-20V:	border supply (floating) @ .1A	(2W)

The total expected maximum load is approximately 150W. The size is expected to be approximately 2" x 10" x 6" and the weight 3 lbs.

The AC main voltage is first rectified and filtered through a relatively small filter capacitor. The result is a 100 to 170 volt unregulated voltage source. From this source, a switching regulator exhibiting approximately 80% efficiency is used to generate a well-regulated 60V supply voltage. The 60V power supply is fed to several

small chopping type supplies which in turn generate the required voltages and provide the line isolation required. The efficiency of the secondary regulators/converters is expected to be about 80%.

The total package and final converter design remain to be completed and must be specifically tailored to the terminal requirements.

D. Bitzer
P. Tucker
L. Hedges
D. Hartman

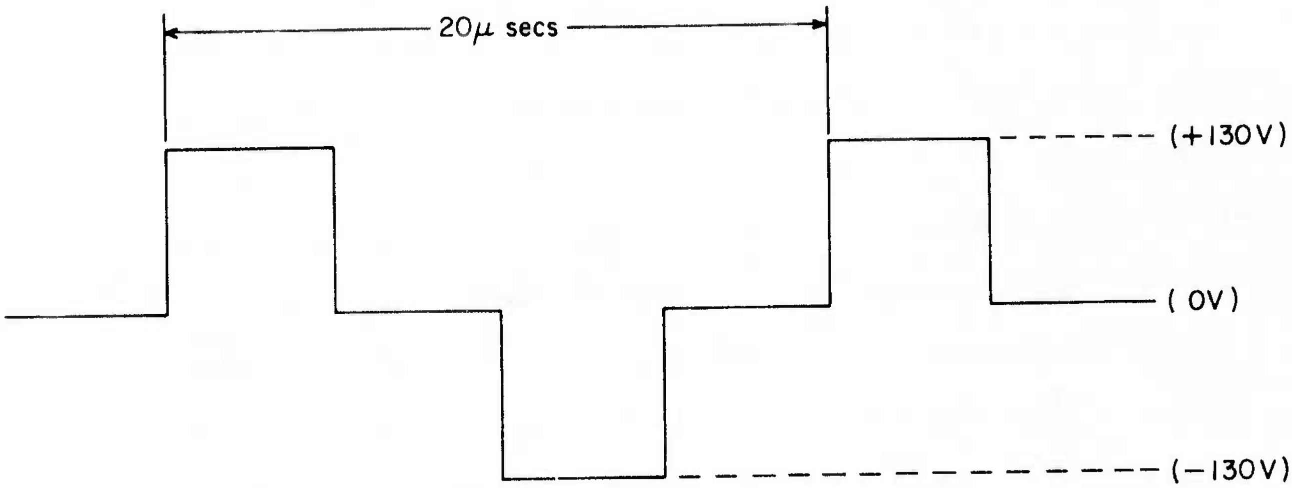
4. NEW SUSTAINER WAVEFORMS FOR PLASMA DISPLAYS

A Plasma Display is a dynamically sustained device. This operation requires the continuous application of a high voltage (100-140V) signal which provides the stimulation necessary to maintain a stable panel state (i.e., those cells which are "on" should stay on, and those cells which are "off" should stay off). In a static sense, the applied waveform must be sufficient to keep already fired cells on and yet insufficient to cause off cells to fire. Figure 4.1 shows a typical sustainer waveform for a nominal PbO plasma panel. This waveform provides a static operating range of approximately 12V (static range is: $V_{on} - V_{off}$ where V_{on} is the lowest voltage at which cells spontaneously turn on and V_{off} is the highest voltage at which cells spontaneously extinguish) on a new plasma device. As a plasma panel is operated, the margin is reduced by nonuniformities in the aging process from cell to cell.

A new sustainer waveform has been developed and is presently being studied. This sustainer is expected to provide much improved operating margins over the classical sustainer. Preliminary studies have indicated an improvement of as much as 250%.

The fundamental concept being explored is the addition of an extra stimulus to the sustainer which provides a mechanism for restimulating partially mature discharges of the type found in cells on the verge of extinguishing. This mechanism causes a dramatic reduction in the V_{off} potential and thus broadens the operating range.

D. Bitzer
P. Tucker
L. Hedges
D. Hartman



TYPICAL PLASMA DISPLAY SUSTAINER

FIGURE 4.1

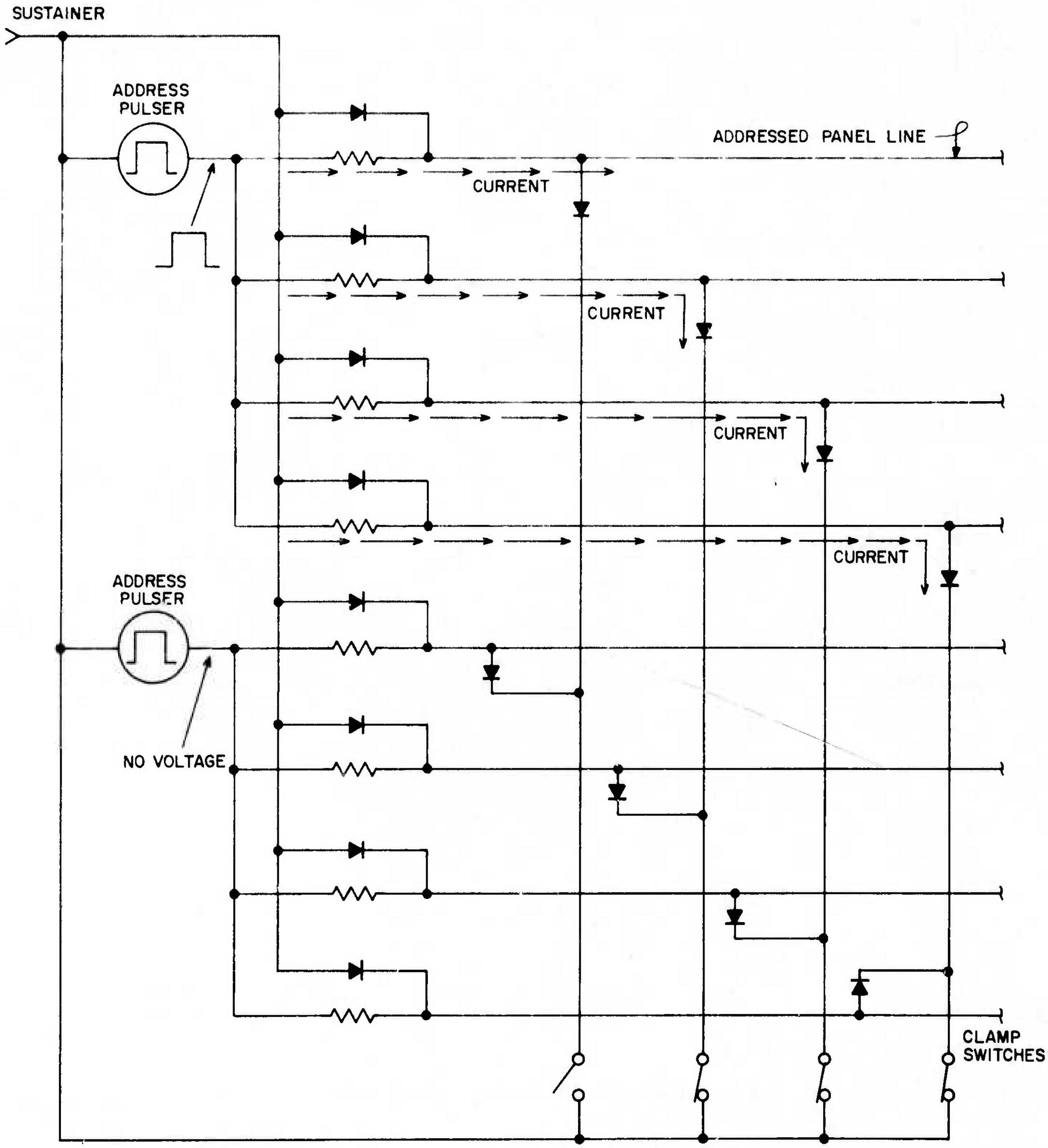
5. NEW PLASMA DISPLAY DECODING TECHNIQUES

A major component of the plasma display system is the decoding and addressing circuitry. This circuitry, which must provide a unique voltage waveform to each panel line (1024 lines on a 512 x 512 panel), is complicated and costly. Contemporary display decoders/drivers involve three components per line -- one resistor and two diodes. These three components comprise a two input gate so that voltage at one panel line is the result of two valid gate inputs. Figure 5.1 demonstrates this technique.

The new driver/decoder circuitry being developed involves just two diodes per line arranged in what could be termed a time-sequenced, two input AND gate. There are two criteria which must be satisfied at a particular cell before addressing (either write or erase) can take place: sufficient voltage must be applied and it must be applied for a sufficient duration as well. In previous forms of circuitry, only the first phenomenon has been exploited by applying half of an adequate address potential on one panel axis and half on the other axis. The result was that only at the one cell which was a member of both axis lines was fully adequate voltage present. The new technique being studied involves the application of an adequate voltage to several lines on each axis but applying it for a sufficient period of time as well as with a sufficient amplitude on only one line of each axis. Whereas in the classical system there were three different possibilities of cell voltage -- fully selected, half selected, and non-selected (figure 5.2) -- the new time-sequenced mechanism provides a total of six different cell waveforms --

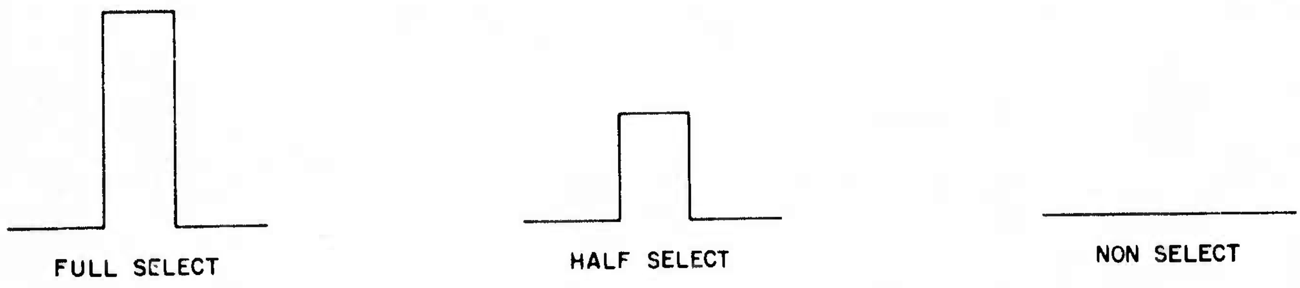
fully selected, true half selected, half select plus time half select, full time half select, time half select, and non-selected (figure 5.3). However, there is still just one adequate condition, that being full select.

D. Bitzer
P. Tucker
L. Hedges
D. Hartman



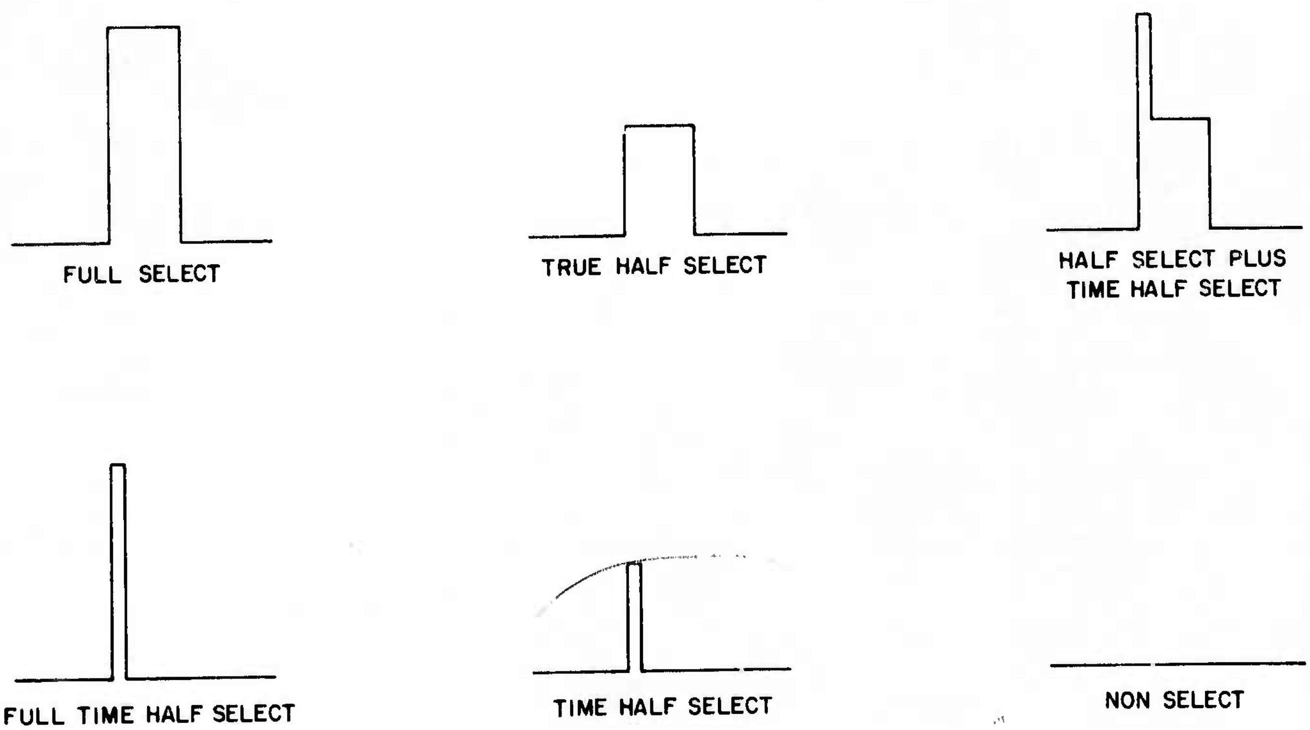
TYPICAL DIODE/DIODE/RESISTOR DECODE/DRIVER

FIGURE 5.1



DIODE/DIODE/RESISTOR ADDRESS WAVEFORMS

FIGURE 5.2



TIME SEQUENCED ADDRESSING WAVEFORMS

FIGURE 5.3

6. PLATO TERMINAL DEVELOPMENT

Two additional terminal processing units have been constructed. One will be used to operate a high resolution (1000 line) cathode ray tube (CRT) display while the other will operate a new version of the plasma display unit now being developed. The CRT terminal will be used to evaluate the quality of PLATO displays on such a device. The new plasma display unit will employ new addressing and panel driving circuits which are described elsewhere in this report. Successful realization of these techniques will result in a thin (approximately 2 inches thick) plasma display package. A significant cost reduction should also be realized.

The terminal processor has been modified slightly to permit NOP codes (words containing all zeros) to enter the processor. Previously these codes, which are automatically generated by the NIU when no data is present, were detected by the serial input register and discarded with no interrupt generated to the processor. These codes now enter the processor but no action is taken on them. This change was implemented to permit PLATO (software) generated NOP codes (same as an NIU NOP with a "one" in the least significant bit) to be sent to the terminal and which will be counted as words received but no other action taken. These PLATO NOP codes can then be used as 1/60 second time fillers to prevent the PLATO system from sending information to the terminal faster than it can be processed. This situation can occur because the prototype terminal can perform tasks that may take longer than 1/60 second to complete. (The existing PLATO terminal always finishes processing data in less than 1/60 second.)

The PLATO operating system has been modified to incorporate the block erase feature as it is implemented in the prototype terminals. This change permits existing TUTOR programs containing an "erase" command to perform the erase in a block mode rather than as a character by character erase. PLATO automatically interrogates the terminal to determine which form of the erase is to be used.

The boldface character set has been discarded in favor of a terminal resident algorithm which will perform a 2X magnification of the characters stored in the terminal. Data in any of the character memories be magnified. The number of character memories (64 characters) has also been increased from four to eight.

A real time clock has been added to the terminal resident. This clock keeps a record of time in 1/60 second intervals and is available for use by any terminal resident program.

J. Stifle
M. Hightower
L. Hedges

7. AUDIO VISUAL FACILITY

7.1 RANDOM ACCESS AUDIO

Education and Information Systems, Inc., the successful bidder for random-access audio devices, delivered the first ten of the 102 audio devices ordered. These units are attractively designed, show a large improvement in audio quality compared to the prototype audio devices, especially with respect to dropout, and exhibit good reliability and performance in the field. With the successful introduction of these audio units, CERL's role in audio device fabrication has been terminated. Audio discs for the older audio device models are still being produced at CERL.

There are a number of instructional courses utilizing random-access audio devices in which both the device request rate and service time are quite small. For these cases a shared-audio system in which m students share n audio devices (m>n) may be justified if the additional cost of the control system is much less than the savings in reduced audio units.

An audio-sharing system in which twelve students share three devices is under development and is the subject for a Master's thesis. The control system uses the Intel 4040 microprocessor for receiving and interpreting student requests, assigning available audio units to requesting students or sending back busy signals if that is the case. One of the main purposes of this project is to reduce the costs for random-access audio use. At the time of initial development, the 4040 microprocessor and its associated family of chips were the least expensive for generating the desired control functions.

7.2 RANDOM ACCESS SLIDE SELECTORS

A program for modifying the earlier slide selector models in order to upgrade their performance has been initiated. In these earlier models the maintenance of focus for all images of the microfiche was very difficult. Also, certain alignment procedures were extremely cumbersome. The modifications with respect to both these problems have upgraded their performance to that of the later models.

A slide selector maintenance manual has been printed and distributed to all PLATO terminal sites. The reference is "Random-Access Slide Selector User's Manual", D. Skaperdas and F. Propst, Informal Report, August 1975, CERL, University of Illinois, Urbana, Illinois.

7.3 MICROFICHE PRODUCTION

A semi-automatic color processing system, which can process approximately 40 microfiche in two hours, has been put into operation. This will decrease microfiche turnaround time and possibly improve the quality of color reproduction. Although the system is now in production with results as good as those obtained from commercial sources, tests are being conducted to determine optimum performance parameters for color.

An improved step and repeat microfiche production camera was designed, fabricated and is undergoing tests. This camera features an El Nikkor f/3.5, 63 mm lens. Unlike the older cameras in which the lens and its field stop are moved against the microfiche film for each image exposure, the lens and its field stop in this model are stationary while the film is pushed against the field stop. This feature should enable much better focus control. Additionally, critical adjustments for focus

and format can be done more readily. The new design, based on much experience with the older cameras, incorporates features which should greatly improve the reliability. Initial test results are encouraging.

D. Skaperdas
D. Stolarski
P. Stolarski
L. Streff
G. Traynor

8. ADVANCED TERMINAL TECHNOLOGY RESEARCH

The principal objective of this component of the program is to devise human interface, terminal architecture, and system structure specifications for improving the cost-performance characteristics of computer-based education systems. Work is being carried out in five major project areas; these are:

1. Intelligent Terminal Architecture
2. Display Data Integration
3. Audio Input and Output
4. Manual Data Manipulation (Touch Input)
5. Terminal-based Mass Memory
6. Network Concepts

8.1 INTELLIGENT TERMINAL ARCHITECTURE

The objective of the terminal architecture project is two-fold. First, research is being performed to devise specifications for the design of the hardware, firmware and software for future processor-based PLATO terminals. Emphasis is being placed on the improvement of human factors and cost characteristics. Second, research is being conducted in order to evolve a powerful (yet low cost) laboratory and office terminal design which is compatible with PLATO and which exhibits wide-ranging multi-host and stand-alone capability. Both projects are being carried out using mini- and micro-computer-based plasma display terminals developed at CERL.

The primary results for this period relate to the continuing evolution of the PDP-11/05-based intelligent terminal as a vehicle for testing new concepts in human factors and data structures. During this

period hardware and software documentation for the current terminal system was completed (1). Other significant results included the completion of an intelligent terminal operating system based upon DEC's RT-11 (2); the completion of a study on data compression techniques; and the initiation of a design study for an improved operating system which will support interactive graphic communication with touch input.

8.1.1 AN RT-11-BASED OPERATING SYSTEM

Throughout the period work has proceeded in cooperation with Regional Health Resources Center (RHRC) (2) on combining terminal simulation software with a DEC-issued operating system RT-11. The purpose of this work was to provide a more flexible system on which the graphic communication and image trapping work could be performed. A version of the operating system has been completed which runs the terminal simulation software under RT-11.

8.1.2 APPLICATION OF DATA COMPRESSION TECHNIQUES TO THE PLATO IV COMMUNICATION SYSTEM (3)

With the rapid evolution of various mini- and micro-processor based terminal designs (4), it was considered important to reexamine the communication scheme between the central computer and the terminal in the PLATO system. A study was carried out to determine the effects of various data compression methods from the viewpoint of "central computer-to-terminal" communications on large graphics-oriented timesharing systems such as PLATO IV. The desired goal was to find ways of increasing terminal display speed without increasing the transmission error rate.

The results of the study show that while some special cases can be improved by modifying the currently used method for text transmission, a completely new coding scheme would need to be employed to

achieve any insignificant increase in average transmission rate. For example, using a variable length code a minimum increase of 50% over current display speeds could be achieved; however, it was shown unlikely that such a code could do more than double the display rate.

Of significant interest are the results which indicate that a combination of word lists and Huffman coding could be used to obtain greater compression. Also, suggestions for improving "burst" display speed are given.

The work was performed by Maureen Stone for her MSEE and is presented in detail in Appendix A.

8.1.3 IMAGE STUDIES

During this period an attempt was made to formulate concepts related to images and their use. These activities are described in rough chronological order.

This work began with a general review of the PLATO project. The objectives of this review were:

1) classification of the various functions
2) identification of alternate devices which these functions could perform

The alternate devices considered were 1) local processors and 2) memories attached either to a particular terminal or shared by several. This local sharing was called a "sub-site".

During the course of this review PLATO's evolutionary direction was seen to be a significant factor. This direction was towards a general purpose communication system which includes CAI as one component. Communication was seen as a generalization of instruction. Such a more general use had significant implications in examining the disposition of functions.

Recognition of the computer as a communications medium motivated an examination of human communication. This examination was directed towards identifying those features of messages and symbols that could be enhanced by use of the computer. The structural similarity between the thing being represented and the symbol or message referring to it is such a feature.

The symbols and messages presented on the screen were termed "images" and a simple classification scheme devised for them. A formalism underlying the notion of image was devised which allows one to precisely specify the degree of similarity between various collections of symbols. From an operational point of view it allows images to be combined and transformed using a small class of operators. These operators include an addition, subtraction, translation, magnification, rotation and multiple copying. Multiple copying includes animation of an image as a special case.

During this period an attempt to implement an appropriate communications sub-system on this PDP-11 based intelligent terminal was initiated. The organization of images into files under the RT-11 operating system was completed during this period. Images generated centrally can now be stored locally and reused.

In conjunction with image filing and retrieving, an image creating system was designed but not yet implemented. This system attempts to integrate both a local processor and remote processors. Images are treated like film strips. They can be cut, recombined, and replayed when desired. These operations require additional processing power and memory not currently available to the user. This power and memory would be interpreted by PLATO or some other large central system like a DEC-10.

8.2 DISPLAY DATA INTEGRATION

The primary objective of the display-data integration project is to enhance the performance and minimize the cost of the display systems which are to be specified for future PLATO terminals. The principal results from the last period include 1) an improved understanding of the plasma display device physics whichinfluences memory margin and 2) a display/memory organization design which will allow for non-destructive cursor operations and hard copy.

The work on plasma display device physics is closely coupled with additional work being carried out at CSL. A detailed description of this work and the results is presented in "Discharge Dynamics of the AC Plasma Display Panel" a technical report and Ph.D thesis by L. F. Weber (5).

The work on non-destructive cursor operation and secondary image storage is an extension of earlier work supported by RADC. A report describing this work was released in (6).

The objective of this activity is to lower the cost of providing cursor and hard copy facilities as part of the PLATO terminal. The design concepts will be tested and evaluated during the next period.

8.3 AUDIO INPUT AND OUTPUT

The objective of the audio input and output effort has been two-fold. First, research is being performed to devise techniques for improving the storage capacity and lowering the cost of using pre-stored audio in the PLATO environment. Second, research has been conducted in the areas of electronic voice input and speech synthesis in order to evaluate the impact of these approaches on PLATO requirements.

8.3.1 PRE-STORED AUDIO OUTPUT TECHNIQUES

Work is continuing on the design and construction of a prototype optical disk audio storage system. The experimental recording playback apparatus and the electronic controller are nearing completion. Evaluation of several recording and playback schemes will be carried out during the next period.

8.3.2 VOICE SYNTHESIS TECHNIQUES

The electronic voice synthesis research, previously carried out at CERL, is now being performed at the University of Arizona under the direction of Dr. James Parry.

8.3.3 VOICE INPUT TECHNIQUES

Previous ARPA reports, prepared by CERL, described speech recognition research which was aimed at providing a system which could recognize isolated words reliably from multiple speakers, which would be farily inexpensive, and which would be compatible with current and anticipated PLATO architectures. A technique developed by J. Parry was recently investigated by James Oppenheimer. The results of this investigation are reported in a technical document (MSEE Thesis) entitled "An Evaluation of Certain Voice Signal Characterization Techniques for a Low Bandwidth Speech Recognition System." This report is attached as Appendix B.

The overall conclusion drawn by Oppenheimer upon examining the results of the performance evaluation tests was that the Parry voice input device, at the current level of development, was practical only with short, highly differentiated vocabularies or in special purpose applications (i.e., as a prosthesis device for the handicapped). For individual speakers the reliability of recognition of the digits

(critical to any CAI system) approaches an acceptable level of 80 to 90% proper recognition. Nevertheless, with these rates a correction or conformation system would probably be necessary which might prove quite burdensome to the user. Children would have particular trouble with such a system since they tend to speak somewhat inconsistently and would be quite confused by errors in recognition. See Appendix B for a more detailed discussion of this work.

8.4 MANUAL DATA MANIPULATION

Direct touch input coupled with graphic display capability is proving to be a powerful technique for coupling untrained users to computers. The objective of this research activity has been to determine optimum touch input resolution required in future PLATO terminals.

The prototype high resolution (256 x 256) touch input system described in detail in the previous ARPA progress report has been completed and is in operation on the CERL intelligent terminal. The unit works well. Software is now being developed to support upcoming evaluation experiments. A detailed report of this work is in preparation and will be presented in the next report.

In response to requests for flight training games and simulation, the group is also evaluating a multiple joy stick system for use on the intelligent terminal. If a satisfactory approach is developed, it is anticipated that the intelligent terminal will be used to accommodate PLATO-based flight training and testing exercises.

8.5 TERMINAL-BASED MASS MEMORY

The objective of this research is to devise techniques for realizing a low cost 10^{10} bit local mass store using recent advances in

video disk technology. At present we are carrying on discussions concerning a cooperative research program with a corporation that is a potential manufacturer of commercial video disks. We plan to begin work about mid-year.

8.6 NETWORK CONCEPTS

The objective of this activity is to explore the possible uses of computed networking concepts in the PLATO system environment.

Efforts to establish a communications link between the PLATO network and the existing campus computer service were initiated during this period. An overall strategy of implementation was devised which consisted of establishing 2400 band link between a CDC 3266 communications controller and one of several Remote Job Entry (RJE) stations on campus. Installation is expected in early March.

As an intermediate step a PLATO IV terminal was interfaced to the RJE station. This arrangement allows data being sent to terminal to simultaneously be sent to RJE station. This step was facilitated by the fact that the RJE station is made from a PDP-11. This allowed use of existing intelligent terminal hardware.

R. Johnson
P. Lamprinos
M. Stone
P. Van Arsdall
T. Little
K. Gorey
D. Sleator

9. 9600 MODEM DEVELOPMENT

Development of the 9600 bit per sec modem described in the previous semiannual report has continued. The transmitter-multiplexer has been constructed and preliminary testing has been completed. The receiver demultiplexer has also been constructed and the transmitter receiver pair is being tested.

D. Bitzer
P. Tucker
B. Keasler

10. OPERATIONS

The PLATO IV Operations Group has responsibilities in the following areas: installation, maintenance, microwave, data line communications, and supervision and technical support for demonstrations.

10.1 INSTALLATIONS

During the period covered by this report, equipment was moved from Rand Corporation to the University of Southern California at Los Angeles. Also the site at Aberdeen Proving Grounds was closed down. The following ARPA sites were added to the PLATO IV network: Air Force Academy (9/17/75), Redstone Arsenal (9/15/75), and Ft. Eustis (9/12/75). Information at this time would indicate that further reassignments will be occurring in the near future.

10.2 MAINTENANCE

The maintenance operations consist of two separate but interconnecting areas: the physical repairing of non-working terminals and the repairing of the parts that have been replaced. The diagnosis of a particular problem is either done by personnel at the site in consultation with engineers at CERL. This interexchnage of information, either by terminal or telephone, has proven to be a valuable means of reducing the down time of equipment as well as improving the ability of on-site personnel to do their own troubleshooting. This has meant that the physical repairing of terminals can be accomplished by sending replacement parts to the site, where physical replacement is then made.

As was reported earlier, changes were made in the repair program in order to decrease the amount of time needed to fill out repair

reports. As shown in Figure 1, the checklist and repair comments were interexchanged in the display. The checklist must now have items toggled before a change of status or a repair comment can be entered; however, no backing up or replotting is necessary. This simple change resulted in a 50% decrease in the amount of time required to enter a repair report while at the same time it insured that the checklist (which serves as a control for counts) was always activated.

Figures 2 and 3 show the orderly progression of the reporting process. Figure 2 shows the toggling of the checklist and the instructional changes for repair personnel. Figure 3 shows the change of staus and the preliminary repair comments. The program was also changed to allow for a search according to the RIN number, checklist item, or terminal number as shown in Figure 4. Finally, the status of terminals that are down at any particular time is shown in Figure 5.

Table A shows an analysis of the repair program for the last reporting period. It also shows the number of repair reports and trips made by the ARPA sites. When examining the table, one should be aware that a typical time for shipping a part and installing same is nine days and a terminal is considered down (according to PLATO Operations people) from the time it is reported in repair until it is verified as operational by someone at the site. The down times, therefore, include the time to ship and install the defective part. Table A also shows higher down times for those sites where no personnel are available for troubleshooting and repair. The second greatest time builder is time required when it is necessary to send a man to a site for repairs. Finally, telephone line problems on weekends and lack of available part replacements add to the down time for terminals.

The lack of available plasma panels for replacement purposes was one of the primary reasons for the increase in down time of terminals as compared to the last reporting period. As an example, three panel problems in San Diego required a total of 75 days to repair due to lack of parts.

10.3 MICROWAVE SYSTEM

On December 20, 1975, the PLATO system was shut down so that modifications could be made to the computer power and cooling systems. These modifications will allow the addition of a second Central Processing Unit for increased system efficiency. The modifications were completed and the system was running again on January 10, 1976.

During the down period, a 40-foot tower was erected on the existing microwave tower. The increased microwave antenna height improved the signal to noise ration by 6db at Chanute Air Force Base.

The microwave system performance was further improved by replacing all of the earlier model video receivers with a new low-noise design receiver. Because of these changes, the error rates have been reduced by one order of magnitude.

10.4 ERROR MESSAGES

An error message playback system was incorporated to inform PLATO users of the status of the system during a computer failure. The system uses a broadcast quality tape cartridge machine and cartridges with pre-recorded error messages that state the nature of the problem and approximate repair time.

10.5 REMOTE SITES FIELD SERVICE

The PLATO telephone line analyzer described in the last quarterly report has been completed and is now in use by our field service technicians. A small test circuit has also been developed and will be added to all Novation 202 data modems. This circuit will give an indication of forward carrier, forward data, reverse carrier, and reverse data, allowing a non-technical operator to categorize any transmission faults (see Figure 6).

As the system got progressively larger, it became apparent that terminals must be operated remotely from parent 4800 modem-mux units. Originally this was accomplished by use of independent 202 modems, but this presented a problem for people from the Maintenance Group when dealing with untrained personnel at some of the remote sites. Thus to make card swapping easier for site personnel and to consolidate existing physical equipment, a housing was designed which holds four remote modems driven from a common power supply.

G. Burr
J. Knoke
M. Williams

```
  New report from Remote site                                r.n =
Terminal 346                                                 13603
From orlando    Phone - 6465130
Reporter by dr weller                  of mtc site 22 stn. 16
Reporter comments:
touch panel does not work .. makes no response
to touches.   tested it on another terminal and
still no response.

   CHECKLIST
 a) plasma panel       b) panel power         c) logic card
 d) logic power        e) site controller     f) keyset
 g) slide sel.         h) touch panel         i) audio
 j) phone line         k) modem/mux           l) other


Reported on 18 Feb 1976 at  7.86 hours

      Press the corresponding letter to toggle switch.
      An "+" signifies a problem.  Press NEXT when done.
      Press LAB to go on to next report.
        »
```

Figure 1

```
 New report from Remote site                                  rin =
Terminal 346                                                  13803
From orlando    Phone - 6465130
Reported by dr weller               of mtc site 22 stn. 16
Reporter comments:
touch panel does not work .. makes no response
to touches.   tested it on another terminal and
still no response.

   CHECKLIST
 a) plasma panel      b) panel power          c) logic card
 d) logic power       e) site controller      f) keyset
 g) slide sel.        h) touch panel    +     i) audio
 j) phone line        k) modem/mux            l) other


Reported on 18 Feb 1976 at  7.86 hours

P   s - alter status                     →DATA1 TO UPDATE REPORT←
R   t - alter terminal number             NEXT  for next report
E   p - alter location                    BACK  for index
S   r - write/rewrite repair comments     HELP1 to delete
S   c - to set problem checklist
```

Figure 2

```
  Incomplete report from Remote site                      rin =
Terminal 346                                              13603
From orlando    Phone - 6465130
Reported by dr weller                of mtc site 22 stn. 16
Reporter comments:
touch panel does not work .. makes no response
to touches.   tested it on another terminal and
still no response.

   CHECKLIST
 a) plasma panel       b) panel power         c) logic card
 d) logic power        e) site controller     f) keyset
 g) slide sel.         h) touch panel    ←    i) audio
 j) phone line         k) modem/mux           l) other
Repair comments:
 dennis weller requested to send back touch panal for repair



Reported on 18 Feb 1976 at  7.86 hours


P   s - alter status                      →DATA1 TO UPDATE REPORT←
R   t - alter terminal number              NEXT  for next report
E   p - alter location                     BACK  for index
S   r - write/rewrite repair comments      HELPi to delete
S   c - to set problem checklist
```

Figure 3

44 Reports Now On File

Press

a to see all current reports

c to see reports only from the Chicago area

u to see reports only from Urbana-Champaign area

o to see reports from other areas

f to see finished reports

k to see complete reports

x to search for RIN

s to search by checklist item

t to search by terminal number

LAB to see current maintenance status

LAB1 to submit a report

DATA1 to go to testor

Figure 4

Status of Terminals Reported Down at 13.21.21 02/18/76

Problem	Remote	Chicago	Urbana
Plasma Panel	2	0	0
Panel Power	0	0	0
Logic	3	0	0
Logic Power	0	0	0
Terminal Elec.	0	0	0
Keyset	1	0	0
Slide Selector	1	0	4
Touch Panel	0	0	0
Audio	0	0	0
Phone Lines	0	0	0
Modem/Mux	0	0	0
Other	2	0	2
Not classified yet	1	3	0
Total Terminals Out (out of 921)	10	3	6

If a terminal has more than one problem, Total Terminals Out may be less than the total number of problems.

Figure 5

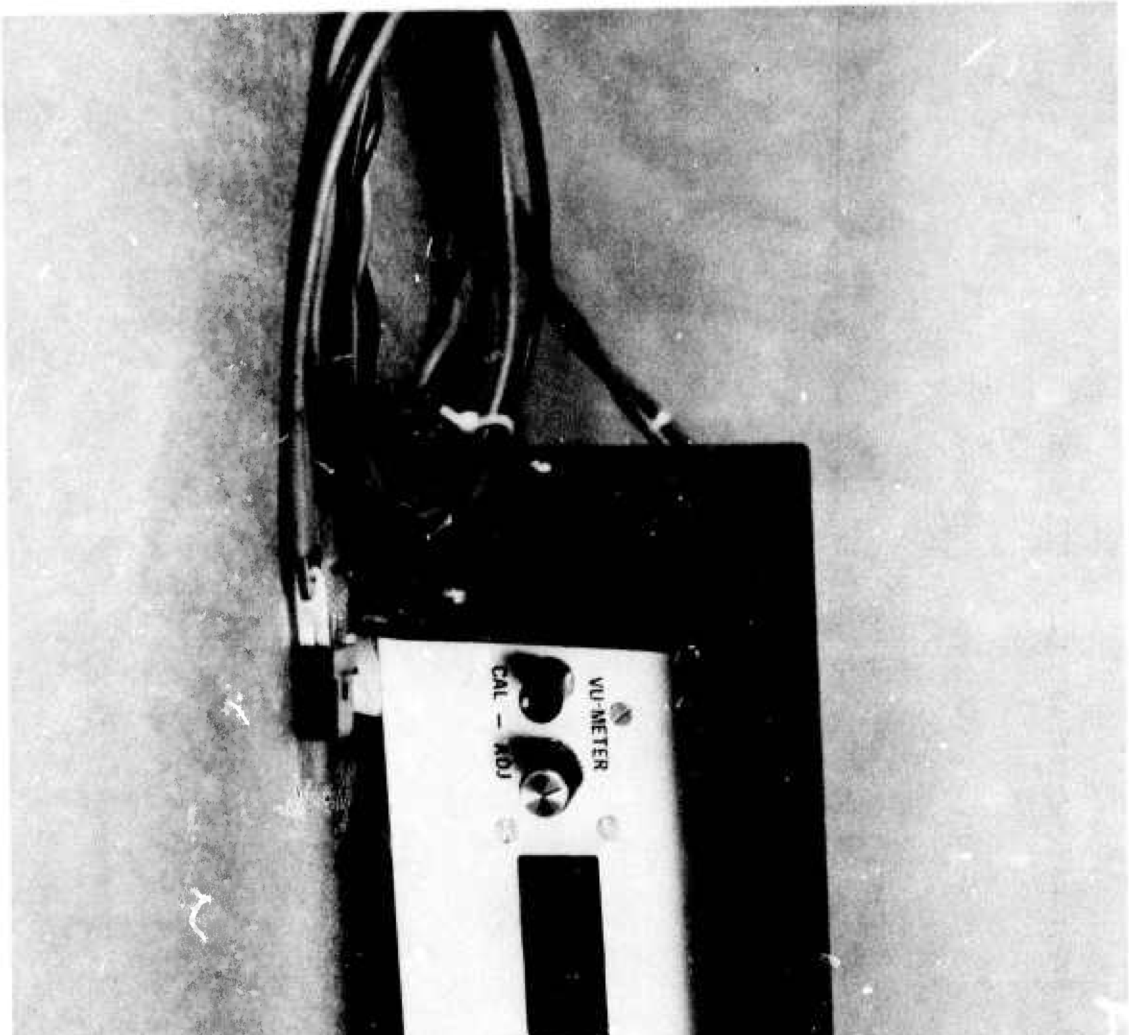


Figure 6

TABLE A

Location	Number of Terminals	Number of Reports	Down Time Term Days	Number of Trips Required	Phone Line Problems
Air Force Academy	2	4	.97	2	0
Army Research Institute	2	2	.00	3	1
ARPA Headquarters[1]	1	3	.86	1	0
Chanute Air Force Base[2]	30	38	71.47	29	0
Educational Testing Service	2	4	72.37	0	0
Electrical Engineering Bldg.[2]	2	5	9.41	6	0
Ft. Belvoir	4	4	51.86	2	0
Ft. Eustis	4	1	0.00	1	2
Ft. Monmouth	1	0	0.00	0	1
HumRRO	2	1	7.14	0	0
Maxwell Air Force Base	4	6	55.37	0	1
Orlando	4	11	93.85	3	9
Redstone Arsenal	4	5	90.35	1	1
San Diego	12	11	163.98	0	6
Sheppard Air Force Base	20	12	104.60	1	17
UC-Santa Barbara[3]	1	0	0.00	0	0
USC	3	0	0.00	0	1
	98	107	722.23	49	39

[1]ARPA and ARI share the same telephone line.

[2]Chanute AFB and Electrical Engineering are shown for comparison purposes.

[3]Santa Barbara is not on line.

Available Terminal Days = 16,273

Percentage Down Time = 4.44%

Note: The system was down from December 20, 1975 - January 10, 1976 for updating of equipment.

A comparison of Table A with the last reporting period is shown in Table B:

TABLE B

	Present	Last
Number of sites	17	16
Number of terminals	98	95
Number of reports	107	123
Number of down days	722.23	462.31
Trips (total)	39	44
Trips (remote)	14	11
Available terminal days	16,273	17,195
Percentage down time	4.44	2.69

The increase in the number of terminal down days was partially explained earlier in this report. The available terminal days decreased because the system was down from December 20, 1975 to January 10, 1976 for updating of physical facilities.

References

1. Intelligent Terminal Documentation, Informal Report, Computer-based Education Research Lab, University of Illinois, Urbana, (October, 1975).

2. Williams, B. T. and R. L. Johnson, Final Report IRMP-48, Illinois Regional Medical Program, (September, 1975).

3. Stone, Maureen C., "Application of Data Compression Techniques to the PLATO IV Communication System," M.S. thesis, University of Illinois, Urbana, (1976).

4. "Demonstration and Evaluation of the PLATO IV Computer-based Education System, (Computer-based Education for a Volunteer Armed Service Personnel Program)," reports for the period August 1, 1972 - January 1, 1974, and January 1, 1974 - January 1, 1975, Computer-based Education Research Lab, University of Illinois, Urbana.

5. Weber, L. F., "Discharge Dynamics of the AC Plasma Display Panel," CSL Report R687, University of Illinois, Urbana.

6. Johnson, R. L. Final Report - Rome Air Development Corp. Contract No. F30602-73-C-0370, Computer-based Education Research Lab, University of Illinois, Urbana, (February, 1975).

APPENDIX A

APPLICATION OF DATA COMPRESSION TECHNIQUES
TO THE PLATO IV COMMUNICATION SYSTEM

Maureen Celinda Stone

TABLE OF CONTENTS

1. INTRODUCTION

With the development of various mini and micro-processor based terminals, it is appropriate to re-examine the question of communication requirements between the central computer and the terminal in the PLATO system. With a processor in the terminal, it is reasonable to reconsider decoding algorithms that were previously too expensive in terms of terminal hardware.

This paper presents a study on the effects of various data compression methods from the viewpoint of central computer to terminal communications on a large graphics oriented timesharing system, PLATO IV. The desired goal is to increase terminal display speed without significant increase in the transmission error rates. While the major emphasis in this paper is on text transmission, some discussion of other display functions is included. The paper is organized into seven chapters.

Chapter 2 provides a general description of the PLATO IV architecture and communications system.

Chapter 3 is a review of two projects involving processor based terminals, one using a 16 bit mini-computer, and one using an 8 bit micro-processor.

Chapter 4 gives a detailed explanation of how text is currently transmitted, followed by an analysis of the average number of bits/character obtained by this method. Three areas for improvement are described.

The theoretical background for variable length or Huffman coding is introduced in Chapter 5. Both the projected gains, and some design considerations for implementation on PLATO IV are given.

The use of word lists is a method successfully used in other applications to obtain a significant reduction in the average number of bits/character used to represent text. Both the projected savings using this method, and the overhead involved are described in Chapter 6.

Chapter 7 contains both conclusions and suggestions for future research. Projects involving text compression, and other methods of improving display speed are discussed.

2. PLATO IV SYSTEM ARCHITECTURE

2.1 Central Computer Architecture

The PLATO IV computer-based education system consists of a large central computer, the Control Data Corporation Cyber 73-24, with more than 900 graphics terminals connected to it. (5,10) The Cyber 73-24 is a dual processor system with the two central processing units (CPU's) connected to the same central memory (Figure 2.1). Two million 60 bit words of extended core storage (ECS) are directly coupled to central memory by high speed block transfers. The ten peripheral processing units (PPU's), which are small, programmable processors, can access both ECS and central memory. Most input/output information is transferred through the PPU's to buffers in ECS. In this way, ECS becomes the central transfer point for all data. (1)

2.2 Communications Architecture

The communication system for the terminals is unusual, as can be seen in Figure 2.2. The data rate is asymmetrical, with the output rate to the terminal being 32 times faster than the input rate. Standard television equipment and voice grade phone lines were used to give the lowest possible cost.

Information for a given terminal is sent from the central computer through a PPU to the Network Interface Unit (NIU). There it is interleaved with the information for all other terminals and transmitted as a video signal. At a particular location, the site controller selects the data for the terminals at the site. The information is

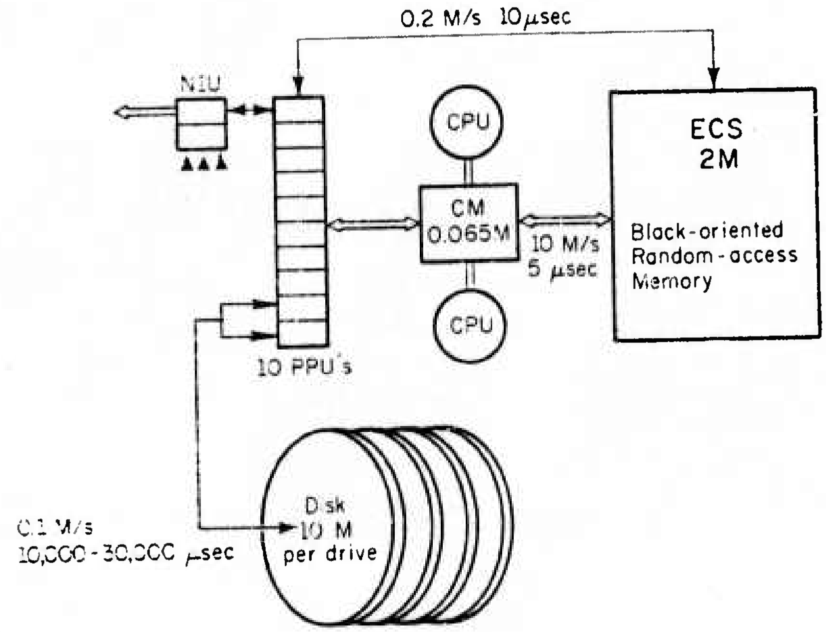


Figure 2.1. PLATO IV computer architecture, showing memory sizes in 60-bit words, transfer rates in 60-bit words/sec, and access times in microseconds. M = million, CPU = central processing unit, PPU = peripheral processing unit, NIU = network interface unit, CM = central memory, ECS = Extended Core Storage. Programs and data are swapped between CM and ECS. Conventional disk drives provide permanent storage for programs and data. The basic computer is a Control Data Corporation Cyber 73-24.

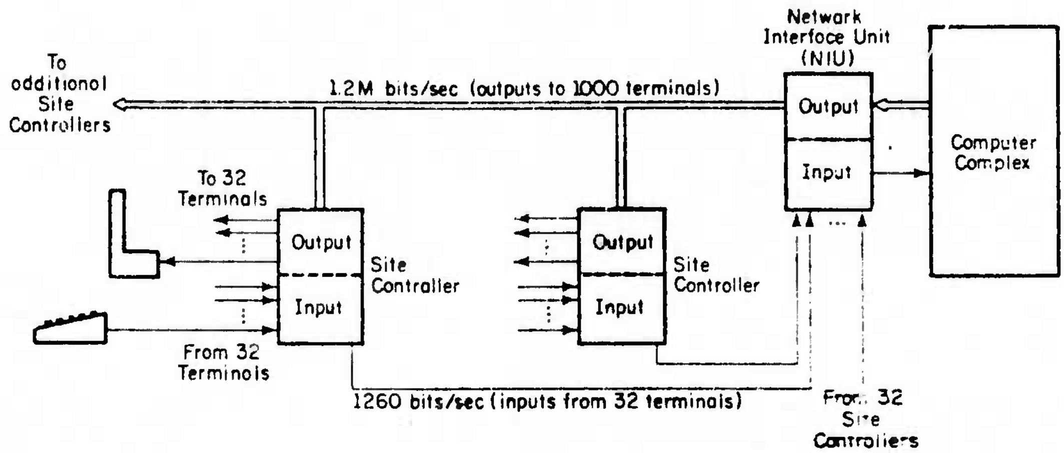


Figure 2.2. Communications hardware configuration.

separated and sent to the appropriate terminal over a voice grade phone line. The limiting channel is the phone line, therefore the data rate to the terminal is usually given as the rate along this phone line, or 1200 baud.

Input, which is usually key presses, is transferred to the site controller along the reverse channel of the phone line. The input data for all terminals at the site is transmitted over a single 1200 baud line back to the central computing system. Since there can be up to 32 terminals at one site, the data rate back is up to 32 times slower than the data rate out. More detailed information on the communications system is given in (1).

All terminals on the PLATO IV system use the same information protocol for output which is unique to the PLATO system. Every 16.7 ms, a 21 bit parcel containing 19 bits of information, 1 bit parity, and 1 start bit, is received by every terminal. This is either information from the central computer or an all zero NOP generated by the site controller. This length format was chosen to accomodate the 9 bits x and 9 bits y needed for panel addressing. An extra bit was needed to distinguish data from control words. Because the system is synchronous, every 16.7 ms a frame must be generated by the central site, consisting of one 20 bit parcel of information for each terminal which has output pending. The output is originally generated by a running program or "lesson" (Figure 2.3). The bulk of the lesson is resident in ECS, with only a small logical block or "unit" resident in central memory. Output is encoded by the Executor

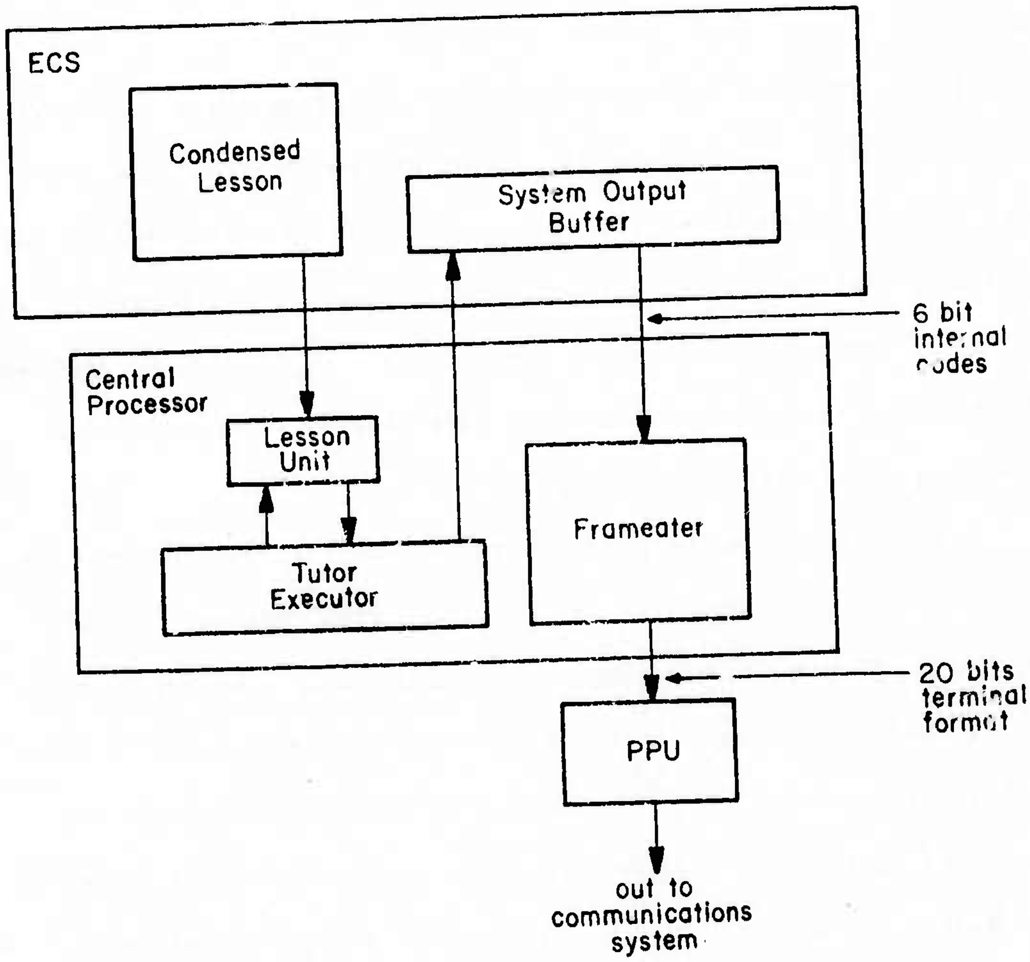


Figure 2.3. PLATO IV output software configuration.

and placed in the system output buffer. However, the information in this buffer is in a generalized, terminal independent form, and not in the 20 bit format required by the PLATO IV terminal. The conversion to terminal format is handled by a separate program, which also periodically creates the frame described above and sends it, through a PPU, to the communications system. This same program, called the Frameater, also keeps track of each terminal's current state to avoid sending redundant information. While 20 bits/parcel are sent by the Frameater to the NIU, parity is actually generated by the communications hardware.

3. REVIEW OF WORK WITH PROGRAMMABLE TERMINALS

The current PLATO IV terminal consists of a 512 x 512 matrix plasma display, keyset, and a touch input device called a touch panel. Available display functions are line drawing, character plotting, and single point plotting. There are 252 available characters, ½ of them dynamically user-programmable from the central computer. Most of the current terminals realize these functions through a MSI/TTL design currently manufactured by Magnavox. (2)

However, it has been recognized throughout the history of PLATO IV that it would be valuable to use a processor in the terminal. During the procurement of the first PLATO IV terminals, a processor-based design was considered, but rejected on the basis of cost (11). More recently, with the evolution of low-cost LSI micro-processor technology, consideration has again been given to processor-based PLATO terminals. This concept has been explored through two projects at CERL.

In 1972, a project directed by R. L. Johnson was started using a Digital Equipment Corporation PDP 11/05 as the basis for a programmable or "intelligent" terminal. Besides the use of the processor, this terminal differed from the standard one because it used a version of the plasma panel which could operate on 16 display points in parallel. This modified panel was therefore capable of a display speed up to 16 times faster than the standard panel. Results of this project are published elsewhere (3).

The most interesting feature of this programmable terminal was the ability to combine high speed display with the flexible presentation

structure of the PLATO IV system. That is, having the PLATO lesson determine the basic design of the display, and having the mini-computer help to get it up on the screen quickly. For example, a major difficulty with display devices such as the plasma panel which have inherent memory is that to erase an area takes as long as it does to write it, with the exception of the full screen erase. For the standard system, due to the synchronized communication and the speed of the plasma panel, area erasure is limited to the maximum character plotting rate of 180, 8 x 16 characters per second. For the programmable system, a terminal function called "block erase" was defined that, given opposing corners of a rectangle, would erase the area. Using the parallel panel, this achieves impressive speeds. Other defined functions for the system include circle generation, rectangular and circular shaded areas, and large sized characters. For more specialized displays, a protocol was defined for loading and calling PDP-11 subroutines from PLATO lessons. Within the PDP-11, system subroutines were available for most display functions. However, it is impossible to match the ease of designing a display as is done on PLATO with subroutines for a mini-computer assembly language. Both the language and the utilities are lacking. But, it is possible to locally store the 20 bit parcels provided by the PLATO generated display, feed them back through the terminal simulator, and see a large increase in display speed. This process, called image trapping, has been successfully used to plot most of the displays in a group of highly interactive medical information system lessons. The major draw-back is the large amount of storage needed. For more than a

few full page displays, it is necessary to use an auxiliary storage medium such as a floppy disk. This project is continuing; expansions of capability include a mini-computer operating system, and advanced peripherals.

In 1974, a project to design a PLATO IV terminal which would combine low cost with expanded terminal capabilities was started under the supervision of J. E. Stifle. Some of the results of the earlier project have been included, and the finished design will be used as a prototype for the next generation of PLATO terminals (4). Several versions of this device, which is based on an INTEL 8080 and a parallel plasma panel, have been completed. The resident system, currently stored in PROM, includes block erase, double sized characters, programmable margins and tabs, and multi-directional text display. 4K of RAM is available for user programs, which can be called from a PLATO lesson. Work is still being done to determine what other features should be part of the standard system and which should be offered as user programs.

4. ANALYSIS OF CURRENT TEXT TRANSMISSION METHODS

4.1 Background for Analysis

For a system such as PLATO IV with a large number of interactive terminals running simultaneously, host-to-terminal communication is a major part of the system load. With the design of the next generation terminal nearing completion, it seemed advantageous to study the overall system format from a communication/information point of view. First, it was necessary to determine the current distribution of display type information. From this distribution, it can be shown that text constitutes the major part of display activity. Therefore, ways to optimize the average number of bits/character sent has been the major emphasis of this project. Starting with a detailed analysis of the current character transmission method, both optimization of the current scheme, and methods requiring more radical changes to the system will be discussed. Both character-by-character and word-by-word compression methods have been considered. However, it has been assumed that no basic changes to the overall communications hardware will be made.

One way of determining the distribution by display type of the information sent to the terminals is to monitor the output of the Frameater or of the PPU (Figure 2.3). At the time it was not practical to put a monitor at either location. The easiest place to sample was at the ECS resident system output buffer. The effect on the output stream could then be deduced. Using this method, one can determine that approximately 50% of all output is characters, 30% screen positioning information, and 20% lines. However, of the 30% screen positioning information, almost

25% of the 30% is taken up by returning to a software set margin. This will be eliminated by the variable set margins, already standard for the new terminals. It therefore seems most profitable to optimize text transmission. A description of the current character encoding methods for the terminal and the central system follows.

4.2 Terminal Character Format

The present PLATO IV terminal recognizes two types of 20 bit parcels or words; control and data. Normally, the Load Mode control word is used to set the terminal mode to either line, character, dot, or load user character memory. All data words that follow are interpreted relative to the mode. Control words include load mode, set x/y, and references to external devices.

The character format for the terminal involves the use of 6 bits packed three to a 20 bit data word. Bit 19 = 1 indicates that the word contains an 18 bit field of data. (Figure 4.1)

19	18 13	12 07	06 01	00
1	CHAR 1	CHAR 2	CHAR 3	P

Figure 4.1. Character Mode Data Word

The 252 possible characters are arranged in 4 memories of 64 characters each. One character position in each memory (o77, where the preceding o indicates an octal number) is defined as an "uncover" code. The combination of an uncover code and another 6 bit code is used to

indicate a change into another memory, or one of several special functions as described in Table 4.1.

To plot characters, the terminal is first set into character mode with a load mode control word. All subsequent data words are interpreted as above. Each character plotted automatically increments x by 8. Note that the carriage return function (o7715) is only useful in the special case where the left margin is at x = 0. To set either x or y, a 20 bit control word must be sent to the terminal. However, this is done without affecting the terminal mode.

4.3 Central System Character Format

Within the central computer system, characters are also kept as 6 bit codes. Since there are 252 characters, plus special functions, combinations of 6 bit codes are necessary. The combinations are rather complex. The code o75, called font, is used as a locking toggle to delineate the alternate font, that is, the user programmable character memory. Within the set of 126 characters of either font, two more special codes are used; shift (o70) and access (o76). The following combinations are possible: 6 bit code; shift + 6 bit code; access + 6 bit code; shift + access + 6 bit code. Therefore, a maximum of 18 bits can be used to designate a character in either font. Other special codes are used to indicate positioning information such as superscript, subscript, etc. A complete list is given in Table 4.2.

This rather awkward encoding scheme is much more reasonable when thought of relative to the key presses needed to create particular characters. The shift code directly relates to upper and lower case on

Table 4.1 Control Functions Following an Uncover (o77) Code

Code	Name	Function
o00	character NOP	no change
o10	backspace	x ← x-8
o11	tab	x ← x+8
o12	line feed	y ← y-16
o14	form feed	x ← 0, y ← 496
o15	carriage return	x ← 0, y ← y-16
o16	superscript	y ← y+5
o17	subscript	y ← y+5
o20	select M0	set to character memory 0
o21	select M1	set to character memory 1
o22	select M2	set to character memory 2
o23	select M3	set to character memory 3

Table 4.2 Special Function Codes for Central Computer Encoding Scheme

Code	Name	Function	Terminal Code
o66	subscript	non-locking $y \leftarrow y-5$ for 1 character then y restored	o77 17, after the character, send o77 16 (unlock)
o67	superscript	non-locking $y \leftarrow y+5$ for 1 character, then y restored	o77 16, after the character, send o77 17 (unlock)
o70	shift	character definition	approximately selects M1 not complete correspondence
o71	margin return (carriage return)	$x \leftarrow 0$ $y \leftarrow y-16$	o77 15
o74	backspace	$x \leftarrow x-8$	o77 10
o75	font	define alternate font	following characters will be in M3 or M4
o76	access	character definition	approximately selects M1 not complete correspondence
o7066	locking subscript	$y \leftarrow y-5$	o77 17
o7067	locking superscript	$y \leftarrow y+5$	o77 16

a typewriter style keyboard. The characters preceded by an access are not visible on the key caps and are mostly mathematical or foreign language symbols. Effort has been made to relate the key to the symbol, such as defining π as access p. While this is the historical basis for the coding scheme, it is not necessary to keep it this way. The elimination of the 18 bit access shift-character combination would considerably simplify character string manipulation, including the translation to output format. No additional overhead would be involved storing input keys, since, for most cases, a translation is already made between the value produced by the keyset and the value described above.

4.4 Description of Text Transmission

Using a 6 bit code for transmission to the terminal has two major advantages. First, 6 bits per character will fit into 18 bits with no overhead. Second, it is possible that an average of less than 8 bits/character, which is the number needed for a straight BCD method, can be obtained because there should be relatively little switching between terminal memories. While certain foreign language and scientific symbols must readily be available in an education oriented system, it is not expected that the average frequency of these symbols will be very high. Therefore, it should be possible to optimize the character transmission rate by carefully distributing the characters among the memories. This can be done by grouping all frequently used characters together, although, what symbols are used in combinations must also be considered. It was decided to place the lower case alphanumerics plus commonly used punctuation and arithmetic symbols together in M0. All

other ROM characters are in M1. These groupings can be seen in Figure 4.2. It was expected that foreign language lessons using a non-Roman alphabet would arrange the characters similarly in M2 and M3.

The following discussion will be based on the results of a system-wide sampling program. Details on this program can be found in Appendix A.2. These particular numbers are taken from an approximately one million character sample taken periodically throughout one afternoon. Although one million characters accounts for less than ten minutes of the total output flow from PLATO IV at such a time, the distributed sampling technique should give an accurate picture of the average situation. While a rigorous analysis has not been done to prove that this is true, several such samples have been taken and are consistent.

The actual character distribution can be seen in Figures 4.3 and 4.4. The space code is by far the most frequent character. In this sample, it represents around 25% of all characters sent, while 20% is considered typical for English text. The difference is partially due to the lack of a multi-character TAB function which requires that space strings be sent instead. Note that the space character appears both in M0 and M1, to avoid memory switching for this common case. After the space, the lower case alphabetic characters follow the normal English distribution.

In this particular sample, several character codes do not appear at all. One of these, the arrow seen at the far right in Figure 4.3 is actually quite prevalent system-wide. However, due to historical reasons, it is not encoded in the same manner as the other characters in the system output buffer, and as such was not seen by the sampling

ADDRESS (OCTAL)	M0 CHAR	M1 CHAR	ADDRESS (OCTAL)	M0 CHAR	M1 CHAR
0	:	#	40	5	↑
1	a	A	41	6	→
2	b	B	42	7	↓
3	c	C	43	8	←
4	d	D	44	9	~
5	e	E	45	+	Σ
6	f	F	46	-	Δ
7	g	G	47	*	∪
10	h	H	50	/	∩
11	i	I	51	(	{
12	j	J	52	)	}
13	k	K	53	$	&
14	l	L	54	=	≠
15	m	M	55	SP	SP
16	n	N	56	,	|
17	o	O	57	.	o
20	p	P	60	÷	≡
21	q	Q	61	[	α
22	r	R	62	]	β
23	s	S	63	%	δ
24	t	T	64	×	λ
25	u	U	65	⇐	μ
26	v	V	66	'	π
27	w	W	67	"	ρ
30	x	X	70	!	σ
31	y	Y	71	;	ω
32	z	Z	72	<	≤
33	0	~	73	>	≥
34	1	¨	74	_	θ
35	2	^	75	?	@
36	3	´	76	≫	\
37	4	`	77	UNCOVER	UNCOVER

Figure 4.2. ROM character memories.

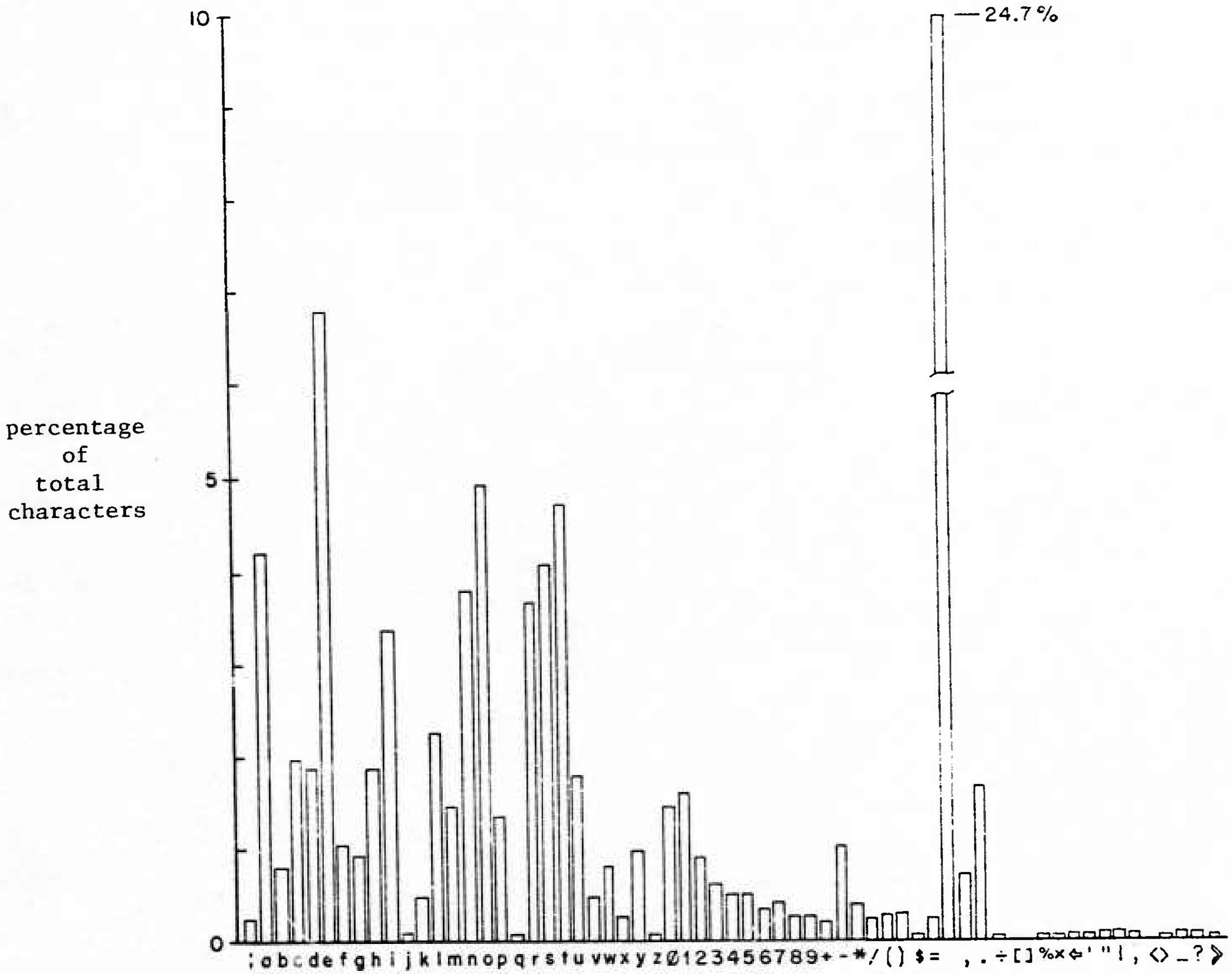


Figure 4.3. Character frequency distribution for MO.

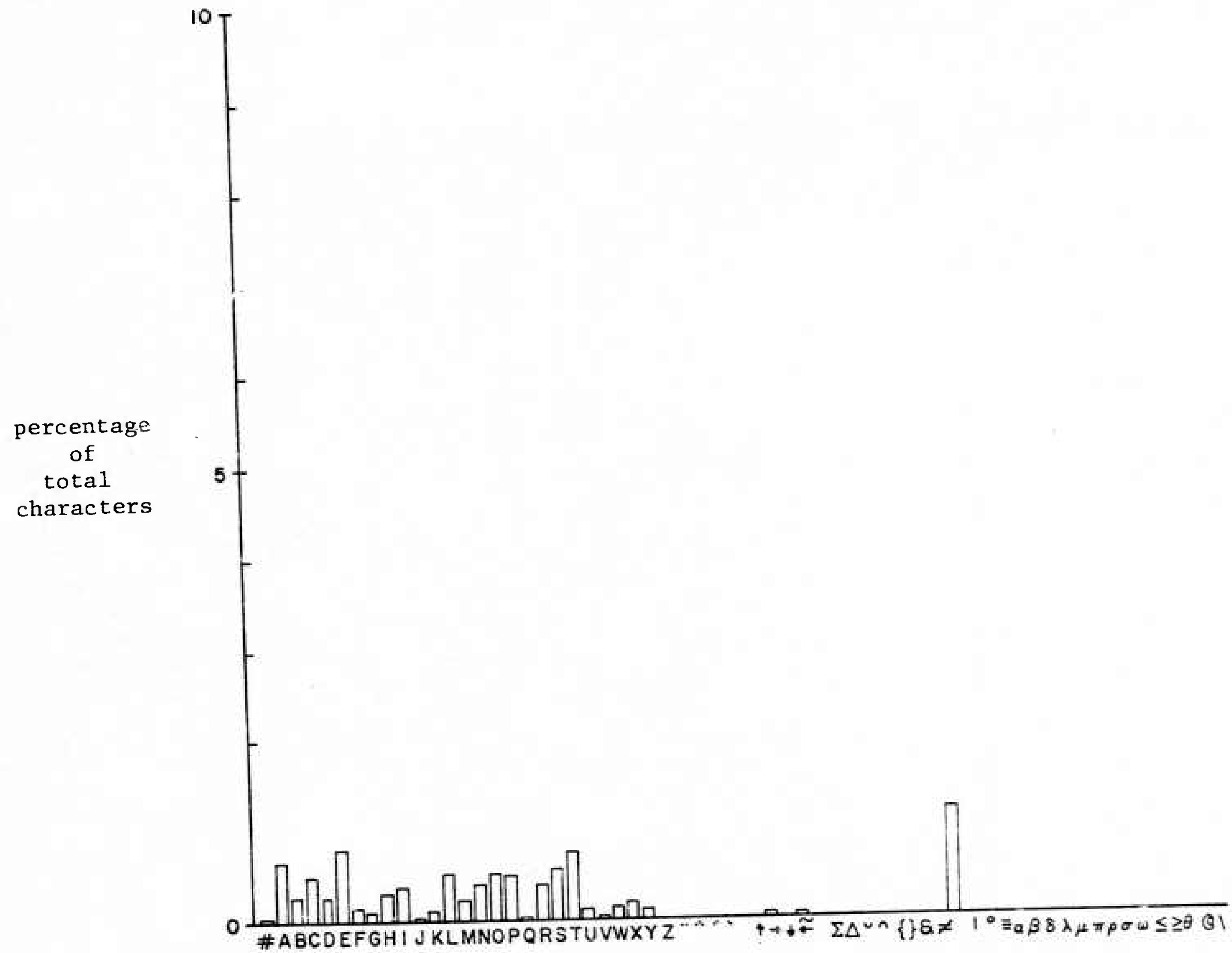


Figure 4.4. Character frequency distribution for M1.

program. Other characters that do not appear can be assumed to be infrequently used by the system as a whole. On inspection, it can be seen that they are either special mathematical symbols or foreign language symbols, which are very dependent on the type of lessons running. The type of lessons running depends on which classes are being taught at the time of the sample. These characters do appear in more selective samples.

Besides the character frequency data, information on individual memory usage and the distribution of memory transitions was taken over the same sample. The results indicate that 88.1% of all characters plotted resided in M0, 8.0% resided in M1, 2.9% in M2, and 0.9% in M3. As was anticipated, M0 is by far the most heavily used.

Inherent in this coding scheme is the assumption that once a change into a memory is made, the next character is more likely to be in the new memory than the old. This assumption can be checked by comparing the number of transitions out of a memory with the number of times the next character was within the same memory. In the case of M0, it is 20 times more likely that the next character is in M0 than in any of the other three memories. For M1, on the other hand, it is only 23% more likely that the next character is in M1 as opposed to anywhere else. Because M1 contains the upper case alphabet, it was suspected that the M0→M1→M0 transition, which would occur for a word beginning with a capital letter, would be quite frequent. Therefore, a special check for this transition was included. It was found that approximately 60% of the transitions between M0 and M1 were encompassed by this case.

This implies that a non-locking shift to M1 in addition to the current locking transition would be beneficial.

From the same data, it can be determined that 90.5% of the time, plotting a character does not require a change of terminal memory. This can be used to compute the average number of bits/character as follows:

$$.905 \times 6 + .095 \times 18 = 7.14 \text{ bits/character}$$

This is indeed better than 8 bits/character, as was predicted. This is not a completely accurate picture, however. Because of the overhead inherent in the 20 bit parcel scheme, the real number is somewhat higher.

First, each character actually requires 6.3 bits, to include the data/control bit. Recomputing gives 7.47 bits/character. Neither the start nor the parity bits are represented as they are not usually included in a discussion of this kind. However, the effects of these bits would be computed similarly. For ease of discussion, a 6 bit character will be assumed for the rest of this chapter unless explicitly stated otherwise. The higher value can always be obtained by multiplying by 6.3/6.0.

Another source of overhead is due to the fact that there are multiple characters in one data word. This can cause unused bits at the end of a character string. Within the current design, there is no 6 bit code which can be used as a NOP, or fill character. Therefore, it is necessary to go to a 12 bit NOP. The extra bits transmitted in this manner account for 12% of all character output. This increases the

average number of bits per character to 8.00. This is the number of bits required by a straight BCD encoding scheme, although it would not be possible to implement such a scheme directly without considerable overhead if the 20 bit parcel size were retained. It seems safe to assume that the use of a 6 bit NOP would reduce the fill overhead to 6%. A 6% overhead gives 7.57 bits/character.

Ignoring the 12% fill overhead for the moment, the result of translating this sampling of the output buffer to the format required by the terminal gives 7.64 as the average number of bits per visible character. The difference between this figure and the 7.14 bits/character given before is due to the function codes included in the output stream. By this it is meant those codes described in Table 4.1, other than those used to change memories. Each code is assumed to take 12 bits. A discussion of the effect of the various types of function codes follows.

The most common single code is the margin return, or carriage return. Alone, it accounts for 0.4% of the character output streams. Because the new terminals will have programmable margins, it is expected that this function will become even more significant.

Taken together, the superscript, subscript, locking superscript, and locking subscript constitute 1.1% of the total character output. While the locking type can be sent with a 12 bit code, to do a non-locking superscript or subscript requires 24 bits. For example, to do a non-locking superscript requires a 12 bit locking superscript code to precede the character, and a 12 bit locking subscript code to follow it. In this sample, the extra overhead caused by not having a 12 bit

unlocking superscript and subscript accounts for 0.4% of the total character output stream. While this number is not very large, for certain types of displays the overhead can be significant. For example, take the equation: $y = x_1^2 + 2x_1x_2 + c_1$. There are 14 visible characters, but the superscripts and subscripts require transmitting 20 more. This decreases the character writing rate to approximately 1/3 of what would be predicted by the 14 visible characters alone. Just using a 12 bit code would double the display rate, which is a visible increase in speed. This type of equation is common in mathematical and scientific lessons. For example, a sample of chemistry lessons showed that the average overhead for superscripts and subscripts was 6%. Furthermore, the locking case was used hardly at all relative to the non-locking case. For the sake of these special cases, a non-locking superscript and subscript function should be considered.

The remaining function codes, with backspace predominant, account for 1.18% of the total character output stream. To summarize, the function codes, assumming 12 bits/code except for the non-locking superscript and subscript which are 24 bits long, are 2.68% of the character output stream. While this number is small, a page of text with a large number of these codes can plot significantly slower because of the relatively large overhead for the code.

4.5 Recommendations for Improvement

Three areas for possible improvement have been identified: the 6 bit as opposed to the 12 bit NOP or fill characters, the non-locking transition from M0 to M1, and the non-locking superscript and subscript.

Below is a description of the effect on the average number of bits/character for each of these. For the rest of this discussion, the value computed using 6.3 bits/character, to include the data/control bit, will be given in parenthesis next to the value using 6 bits/character.

The base figure for comparison is the current average bits/character as computed by the following expression:

$$1.12 \times 6(v + 2(t + f) + 2(usub + usup))/v = 8.55\ (9.0) \text{ bits/visible character}$$

where:

v = number of visible characters in the sample

t = number of memory transitions in the sample

f = number of function codes in the sample

usub = number of unlocking subscripts in the sample

usup = number of unlocking superscripts in the sample

Reducing the fill overhead to 6% gives 8.10 (8.5) bits/character.

Using a 12 bit, non-locking transition for M0→M1→M0, but still assuming 12% fill gives 8.40 (8.84) bits/character. With 6% fill, it reduces to 7.96 (8.36) bits/character.

Only improving the non-locking superscript and subscript transmission gives 8.52 (8.95) bits/character. As discussed previously, the effect of this on the average is slight.

Implementing all three optimizations gives 8.06 (8.50) bits/character. This is an overall savings of ½ bit per character. While this is only a 5.6% increase in display speed, none of these things should be

particularly difficult to implement. As previously pointed out, using a non-locking superscript and subscript could give a visible speed increase in some situations. The 6 bit NOP would require the loss of a character code. However, the eliminated character could be retained through a 12 bit control function, or the number of memories could be expanded. How many characters can be stored will eventually be limited by the cost of the hardware.

5. VARIABLE LENGTH CODING

5.1 Introduction and Description of Basic Principles

The previous chapter has given an analysis of the current status of character transmission in PLATO IV, and listed three areas of possible improvement. All together, the average increase in transmission rate would be only 6.0%, however. To obtain a more significant increase in transmission rate, and thus display speed, it is necessary to look at more sophisticated methods of compression. In this chapter, a definition of variable length or Huffman coding will be presented, followed by a discussion on its applicability to the PLATO IV system. The basic assumption will be that the communications hardware will remain unchanged. That is, transmission will occur synchronously, in 21 bit parcels, 18 bits of which can be character data, and that transmission speed will be limited to 1200 baud by the voice grade phone line.

Within any transmission scheme, there is a finite set of symbols that represent all possible messages sent by the system. The information content for a particular symbol is a function not only of the total number of possible messages, but of the probability of occurence of the symbol itself. An "optimal" encoding scheme is one which transmits no redundant information. To create an optimal code, it is necessary to have the number of bits used by a particular symbol be inversely proportional to the frequency of the symbol. In comparison, most computer character codes use a fixed number of bits/character,

determined by the number of different characters. This method would only be optimal if all characters were equally likely, which is obviously not the case.

A method for creating minimum redundancy, or optimal codes from a set of symbols and their relative frequencies was described by Huffman in 1952 (7). The codes described by Huffman have the following properties: 1) for a given codeword of length L, representing a symbol S, there is no symbol S′ with a codeword length greater than L that occurs more frequently than S; 2) each possible sequence of digits, up to the maximum length codeword, must either be a codeword, or have one of its prefixes as a codeword; 3) no extra information is needed to distinguish each codeword because no codeword is a prefix of a longer codeword.

It is possible to determine the optimal number of bits needed to transmit the information defined by a set of symbols and their relative frequencies. Therefore, it is not necessary to actually construct a minimum redundancy code to compute the savings. The formula is:

average bits/character = H/total # characters in sample

where H is the entropy function defined by:

$$H = \sum_i \frac{f_i}{f_{total}} \log_2\left(\frac{f_{total}}{f_i}\right) \qquad \text{for all } i \in \text{sample}$$

f_i = frequency of i^{th} element

$f_{total} = \sum_i f_i$ = total characters in sample

For the sample used in the previous chapter, this gives 4.95 bits/character. This is 33% shorter than the 7.5 bits/character currently available as the theoretical limit to the PLATO IV coding scheme.

5.2 Implementation on PLATO IV

The implementation of a variable length code on a system like PLATO IV could be done as follows. To encode, a table lookup can be used. This is already done for the current encoding scheme. The characters are then packed into the 18 data bits and transmitted. A fill pattern, such as all 1's, would be used only at the end of text transmission, since character codes can be decoded even if they overlap parcel boundaries.

To decode a variable length code, it is only necessary to consider the character input as a stream of bits. Each bit is examined in turn until a codeword is found. This can then be decoded and the next character started. Since this is a serial operation, it is not necessary to have an integer number of character codes within a parcel. The decoding algorithm can be likened to moving along a binary tree, where each bit determines either a left or right branch. When a leaf is reached, the codeword has been found.

For any new character coding scheme on PLATO IV, care must be taken to include the function codes in the set of transmission symbols. While it is common terminology to refer to the number of characters as 256 (or 252), this is not the case. The actual figure that should be used is 265 for the current system (252 + uncover + 12 functions) and at least 274 for the projected terminal.

6. THE USE OF WORD LISTS OR DICTIONARIES

6.1 Introduction to Dictionary Compression Methods

Up to this point, transmission of text has only been discussed in terms of transmission of a string of character codes. However, the amount of information available in a page of text is not defined only by the information inherent in the individual characters. The organization of these characters into words is also significant. Including this information in a text encoding scheme can be used to drastically reduce the average number of bits per character required. The theoretical limit, as defined experimentally by Shannon in 1951, is 1.3 bits/character (6). Algorithms as efficient as 1.8 bits/character have been defined for computer systems, using dictionaries of words and word by word encoding (8).

The method used is to create a word list or dictionary containing some or all of the words in the text. Each word in the dictionary is assigned an index indicating its position in the list. To encode, this index is substituted for the word in the text. Traditionally, this method has been used to decrease storage requirements, especially for archival storage because to obtain maximum compression requires the use of large dictionaries. Therefore, encoding time, which requires a search through the word list, can be high. However, a study made by Godfred Dewey (9) of printed text indicates that the word "the" alone accounts for more than 7% of all printed text. He also indicates that the first 10 words by frequency account for more than 25%, and the first

100 words account for more than 50% of all printed text. Therefore, it would seem that a significant benefit could be obtained by using a relatively short list of words.

To use dictionaries for host-to-terminal transmission, three areas have to be considered: the distribution of words transmitted by the system, since it is not guaranteed to be the same as that for printed English; the ability of the terminal to decode and plot the received word; and the amount of extra overhead at the central computer caused by the encoding.

6.2 Word Distribution on PLATO IV

To study the word frequency distribution, the program which takes periodic samples from the system output buffer as described in Chapter 4 was used. The sample was then parsed into words and a frequency count for each word was kept. From this list, the impact of dictionaries, on the average, could be deduced. In this program, while the space code was included as a delimiter, some samples were analyzed which also counted space strings as words to predict the benefits of the programmable tab. Further details on the mechanics of this program can be found in section A.3 of the appendix.

The results of this program show that while the frequency distribution is similar to that given for English (9), many of the more frequent words are particular to PLATO IV. Notably, words indicating keys to be pressed, plus the word "press" itself were very common. For one sample of approximately 100,000 words, not including space strings, the most common word was "the", which was 4.6% of all words transmitted. The

first 10 most frequent words include 16.7%, and the first 100 words include 44.3% of all words transmitted. A similar sample, including space strings, gives the double space as the most frequent, at 7.9%, followed by "the" at 2.75%. The first 10 words give 22.2%, and the first 100, 46.0% of all transmitted words.

While the above numbers offer the most direct comparison of PLATO word distribution with other word frequency studies, to determine the effect of a dictionary encoding scheme on transmission speed it is necessary to look at a slightly different measurement. What is needed is the amount of the total output flow that is described by the words. This number is computed as follows:

length x frequency / total characters

length = # characters needed to transmit the word

frequency = frequency of occurence

total characters = total number of characters, including delimiters, transmitted for the entire sample

It was assumed that a space code would be transmitted with the word except in the case of the space strings.

For the sample without the space strings, transmitting the most frequent word, "the", plus a space, defined 3.9% of the total character output. The first 10 words encompassed 14.5%, and the first 100, 38.0% of the transmitted characters. For the sample with the space strings, the results were 8.3% for the first word (double space), 23.4% for the first 10, and 47% for the first 100 words.

6.3 Decoding Algorithms

To decode a dictionary encoded text, it is necessary to know the dictionary, and, if not every word in the text is in the dictionary, to be able to distinguish character codes from word indexes. A simple method compatible with the current method of transmitting characters on PLATO IV would be to have memories similar to M0 and M1, which contain whole words as entries. Words in the "word memories" would then be accessed by selecting the memory with an uncover code, then sending a 6 bit index to select the word. Statistics could be taken to determine whether a locking or unlocking selection would be more efficient. This algorithm, using unlocking transitions, was implemented on the PDP-11 based programmable terminal, and was used to display a sample text with a 30% increase in speed. Unfortunately, to achieve any gains, the words in the memories have to have a transmitted length of greater than 3 characters, as it takes three 6 bit codes to select the word. Most common words are short, so savings obtained by this method would not be very great.

A more efficient variation of this method interleaves characters and words in the same memories. The more common words occur more often than many characters, so the optimal method would be to place the most common words in M0, moving some of the less common letters and symbols in M1. M1 would also contain words as well as letters. The number of new memories needed would then be a function of the number of words added.

Internal to the terminal, the memories would not need to be physically interleaved. Then, however, a translation table would be necessary. This sort of logic could easily be handled by a micro-processor.

Assuming absolute best case, that is, that it takes no more than 6 bits to access a word, the following savings could be obtained. Including space strings, a 10% reduction in output could be obtained with 15 words, a 20% reduction with 52 words, and a 30% reduction with 100 words. Not including space strings requires 26 words for a 10% reduction, 70 words for a 20% reduction, and 130 words for a 30% reduction in text output. These figures were obtained using a formula similar to the previous one:

$$(\text{length} - 1)(\text{frequency}) \ / \ \text{total characters}$$

where the -1 indicates the 6 bits/word needed for transmission.

The previous discussion assumed that the same word list was used for all students. However, the words that are universally common are also short. If the vocabulary were tailored to the lesson, longer words could possibly result in higher savings.

A sample taken from students running organic chemistry lessons was analyzed. The results showed that while the word distribution was distinctly oriented towards organic chemistry, the percent of the characters encompassed by the most frequent words was only slightly higher than for the more general case. For the most common word, CH, the percent savings was 2.19. For the first 10 words, the savings was 10%, and for the first 100, it was 34.4%.

Another specific sample was taken from the system editor. Since the language being displayed is fixed format, the space strings used as tabs were most predominant, followed by those words in the heading

for each page. The first 10 words gives 19.7% of the characters. However, 7 out of the first 10 words are space strings, which could be replaced by a tab function.

There is the additional problem with programmable dictionaries of loading the dictionary. However, this could be accomplished in the same way as loading the programmable characters set. The average number of 6 bit characters per word is around 6.5. Assuming 3 characters every 1/60 of a second, a 100 word dictionary would take less than 5 seconds to load. Up to 17 seconds is needed to load the programmable character set, so a 5 second wait would not be unreasonable.

6.4 Cost of the Encoding Method

It has been shown that approximately a 30% decrease in the information flow, which would correspond to a 43% increase in display speed, could be obtained using a 100 word dictionary. It is also well within the capabilities of the terminal to decode the information. We must now examine the cost of encoding such a scheme.

The optimal place to encode is in the Frameater, since the text string is already being encoded there. The additional overhead for word by word encoding would be the time needed to parse the word, the table storage space, and the time needed for the table lookup. The overhead involved with the table lookup is not excessive. Likewise, for a fixed table for all users, the storage requirement is trivial. However, if user defined tables are used, a separate table for each user must be stored. For a system that runs over 400 terminals

simultaneously, this overhead can be significant, especially since the tables would have to be kept in ECS.

The amount of CPU power that is currently used in formating is conservatively estimated as 1/3 of all PLATO operations. Of this time, the largest part is spent formating text not only because text is the major portion of the output flow, but because the formating process for text is relatively time consuming. Parsing for words would add the overhead of searching for delimiters to each character processed. Under current conditions, the increase in processing time caused by this procedure would degrade system performance enough to completely nullify any gains in display speed obtained by using dictionary encoding.

7 CONCLUSIONS AND FUTURE PROJECTS

7.1 Summary of Results

In this paper, an attempt has been made to show how one might increase character displays on PLATO IV, or a similar system. First, the currently used method was analyzed, and an average rate of 9.0 bits/character was computed for a typical sample. Three areas of improvement were defined which would decrease the bits/character to 8.5, a change of 6%. This implies only a 5.6% increase in display speed.

Second, the limit obtainable using Huffman coding was computed to be 4.95 bits/character for the same sample. As this is calculated without including the overhead generated by end of text fill, or the data/control bit, it is necessary to compare it to 7.5 bits/character, which is the equivalent figure for the optimized version of the current method. This implies an increase in display speed of 50%, or 1½ times faster.

Chapter 6 discussed word list encoding. Using approximately the same 6 bit character based method, as is now used to encode characters, to encode words, a 30% decrease in the volume of text information could be obtained using a 100 word dictionary. This would give a 43% increase in display speed. However, the overhead to encode the words is prohibitive, even for short lists.

In summary, while some special cases can be improved by modifying the currently used method for text transmission, a completely new coding scheme must be constructed to achieve any significant increase in average transmission rate. Using a variable length code will give

a minimum increase of 50% over current display speeds. However, it is unlikely that such a code will do more than double the display rate.

It is possible to work with a combination of word lists and Huffman coding to obtain greater compression. One possible algorithm for this is outlined below. However, for many cases it is not the average rate which is most significant in terms of display esthetics, but the "burst" rate. For example, it often occurs that a complicated display will be transmitted to a terminal, then transmission will stop, or be reduced to a very low level while the user studies the display. Therefore, the average rate of transmission is low, but esthetically the process is slow because of the large amount of time needed to plot the display. Subsequent replots of the display are even more tedious. Suggestions for improving burst display speed for some cases are given in 7.3.

7.2 Suggestions for Future Work in Text Compression

To obtain greater increases than the 50% mentioned above, it would be necessary to go to a combination of methods, such as using Huffman coding with word dictionaries. While this retains the problems of processing overhead, a variation of this might be possible. It was mentioned in Chapter 6 that the double space was a very common pattern. Other two character combinations, which were not analyzed as they were not classified as words by the program, are also common. A coding algorithm using only 1 and 2 character groups would be less expensive than the dictionary lookup, since the Frameater would not have to search

for delimiters. A modified indexing scheme could be used to reduce the search time for valid double character groups. For example, the first character would be used as an index, as it is now, into an encoding table. Each table entry could contain a pointer to a list of double character groups beginning with that character. Thus, a very short table lookup would be the only major overhead. The program which now takes statistics on word frequencies could easily be modified to study this and other multi-character groups.

7.3 Increasing "Burst" Display Speeds

Some experimentation has shown that an increase of average display rate of 20% relative to the current rate of approximately 120 characters/second is scarcely visible. Doubling the rate to 240 characters/second begins to give significant advantages for full screen displays. However, the maximum rate for the parallel plasma panel is near 6000 characters/second. At that rate, it takes only 1/3 of a second to fill the screen. There is no way to use that ability by relying strictly on the average data rate over a 1200 baud line. Even considering the limitations of the 8 bit micro-processor, and using 2000 characters/second as a maximum, this is an order of magnitude more than what was predicted for any of the general text encoding methods. However, it should be possible to use the high speed display in bursts.

One example of such a burst operation is block erase. There, it takes relatively little information sent from the central computer to indicate the rectangular area. Then the local processor can erase the

area at as high a speed as possible, limited only by the local processor and the display. The same principle as block erase can be used for area shading.

This burst capability can be extended to text by storing locally common headings, help sequences, or index pages in a manner similar to the image trapping mentioned in Chapter 3. Also, the user programmable character set is often used to make small, multi-character pictures. After a certain size, it is possible to see the individual characters within the pictures plot. If a translation table were stored locally, indicating which characters fit together, then each figure could be called by a single character code transmitted from the main computer. Especially for characters involved in animations, the improvement in display quality would be considerable.

Another area that can be greatly improved in a burst mode is line drawing. The current method sends an endpoint every 17 msec. For a complicated figure, it may take ½ minute to plot. There are several ways to improve this for special cases. First, it is possible to use image trapping. Second, many line drawings are actually sized characters. Moving the ability to compress and expand character size to the terminal, if possible, would significantly increase the speed of such displays. Other than that, it is necessary to find some method of packing more endpoints in 18 bits of data.

The resolution of the plasma display is 512 x 512, 60 lines/inch. Therefore, it takes 9 bits to give maximum x or y, and 6 bits to describe an inch. One possibility is to pack Δx, Δy, and try to get

three coordinates into 18 bits. As in character strings, it is not essential that whole endpoints arrive in one parcel. However, the decoding operation is not as convenient for such a case here.

Another possibility is to define a larger grid for lines, so that it takes less bits for maximum x and y. Six bit resolution gives a grid of approximately 1/8 of an inch. In fact, there is a commonly used coarse grid already on PLATO IV, corresponding to the character grid, which is 8 x 16 dots. This grid is often also used for lines as well.

A special case can be made for horizontal and vertical lines, such that only one y or x coordinate, respectively, need be indicated. To determine which method would give the greatest gain, it would be necessary to do a sample and analysis program for lines, similar to the one done for characters. An attempt was made to use a modification of the character analysis program to study lines. However, the critical information for line is the distance between endpoints. A strict average would not give the information needed. Therefore, it would be necessary to keep more information as to where the lines are sent to guarantee valid results.

7.4 Elimination of Text Formating

It has been mentioned in 6.4 that approximately 1/3 of PLATO's CPU needs are required for formating. With a processor based terminal, it is possible to eliminate the character formating altogether by accepting the internal codes described in 4.3. As the system gets more processor bound, this becomes an increasingly attractive option. A program to do this has been written for the micro-processor based

terminal, which is basically just a sparce table indexing routine. (12) While a full scale analysis of the internal codes with regards to transmission has not been done, it could easily be performed by modifying the character by character analysis program. Two things would be obvious improvements. First, eliminate the access + shift + 6 bit code characters. This would decrease the decoding table size by 25%. Second, add a lock shift. The relative merits of (approximately) shift and lock shift were discussed in 4.4 with regards to the M0→M1→M0 transition. It was found there that approximately 60% of all shifts are non-locking.

REFERENCES

1. B. Sherwood and J. Stifle, "The PLATO IV Communications System," CERL Report X-44, Computer-based Education Research Laboratory, University of Illinois (1975).

2. J. Stifle, "The PLATO IV Student Terminal," CERL Report X-15, Computer-based Education Research Laboratory, University of Illinois.

3. M. Stone, R. Bloemer, R. Feretich, and R. L. Johnson, "An Intelligent Graphics Terminal with Multi-Host System Compatability," Digest of Papers, CompCon Fall 74, pp. 37-40.

4. J. Stifle, "A Preliminary Report on the PLATO V Terminal," Internal report, Computer-based Education Research Laboratory, University of Illinois, May 19, 1975.

5. S. Smith and B. Sherwood, "Educational Uses of the PLATO Computers System," to be published.

6. C. E. Shannon, "Prediction and Entropy of Printed English," Bell System Tech. J., 30, 50-64 (1951).

7. D. A. Huffman, "A Method for the Construction of Minimum Redundancy Codes," Proc. IRE 40, 1098-1101 (1952).

8. R. D. Cullum, "A Method for the Removal of Redundancy in Printed Text," Thesis, University of Illinois Dept. of Computer Science (1972).

9. D. Godfrey, "Relative Frequency of English Speech Sounds," 1923, Harvard University Press, Cambridge, Massachusetts.

10. D. Bitzer, B. Sherwood, and P. Tenczar, "Computer-based Science Education," CERL Report X-37 (1972), Computer-based Education Research Laboratory, University of Illinois, reprinted in "New Trends in the Utilization of Educational Technology for Science Education," UNESCO, Paris (1974).

11. R. L. Johnson, private communication.

12. B. Sherwood, private communication.

APPENDIX

A.1 Sampling Program

This program periodically samples the system output buffer, screens the information, and places it in a disk file, called a dataset. The parameters for the screening process are: user type, course, lesson, station, and output header code. These items are described below.

There are two main user types, author and student. An author is assumed to be developing lesson material, while a student is studying it. Therefore, the author is often using the editor and other system utilities, while the student will be running under a specific set of lessons. The current average system load is approximately ½ students, and the number is increasing.

Each user is registered in a course. Especially for students, the general area of interest for the user can be determined from this course. For example, students in course chem136a are studying organic chemistry.

The lesson name can be used to define a very specific area of interest, such as the system editor. The station number, which defines a particular terminal, can be used to determine what output is sent to one user, or group of users such as the classroom at the Foreign Language Building.

The format for the system output buffer is a heading, followed by data, repeated. Included in the heading is a code to indicate how the data is to be interpreted. This code is called the output header code, and is used to distinguish characters from other types of output.

The screening parameters are kept in a table which can be edited by a separate program. A sample output, showing data being collected for all chemistry students enrolled in several sections of an organic chemistry curriculum, is given in Figure A.1. Output header codes o002 and o027 indicate text information. This same program can also be used to determine the amount of data sampled as there are five different datasets used to hold samples, each with 126 blocks of 322 words each.

The sampling program is automatically run every hour for a maximum of 10 minutes throughout the day.

A.2 Character-by-character Analysis Program

This program takes the character data stored by the dataset in the sampling program and produces the statistics discussed in Chapters 4 and 5. That is, it is used to determine the character frequency distribution, memory usage and memory transition information, and the data needed to compute the average bits per character sent under various conditions. A page of sample output for all but the character distribution is given in Figure A.2. A brief definition of each term on this page follows. Starting on the left:

PLATO characters: the number of 6 bit internal codes processed for the sample

formatted characters: the number of 6 bit codes sent to the terminal, not including fill

visible characters: number of characters actually displayed. This is the same as summing the frequency distribution for all four memories.

```
flag=run

    Data collecting into dataset stoned2
    block #3, word #321

    Data is from user type student

    Courses        Codes         Stations        Lessons
    ------------------------------------------------------
     chem136a       o002          all              all
     chem136b       o027



To change:>
1.  single entry data
2.  stations
3.  codes
4.  courses
5.  lessons



      Press -BACK1- to update common
```

Figure A.1. Display showing screening parameters for sampling program. In this example, text data is being collected for all students in chem136a and chem136b.

```
plato chars=922075
formatted chars=944167
visible characters=741653
total num.of trans=70542
#of M0→M1→M0 trans=16120

M0   =  653674   88.137%
M1   =   59698    8.049%
M2   =   21434    2.890%
M3   =    6849    0.923%

total =  741655


#shift    =104191  11.300%
#access   = 11998   1.301%
#font     = 19217   2.084%
#lock sup=    781   0.085%
#lock sub=    788   0.085%
#backspace   =  5603    0.608%
#subscript   =  1250    0.136%
#superscript=    666    0.072
#margin return= 19711    2.138%
```

```
#bits/char=7.64 (8.02)
plus 12% fill=8.55 (8.98)

limit by shannon's
bound= 4.96 bits/char


0→0   619263   83.497%
0→1    26636    3.591%
0→2     6052    0.816%
0→3     1727    0.233%
1→0    26739    3.605%
1→1    32883    4.434%
1→2       52    0.007%
1→3       21    0.003%
2→0     5908    0.797%
2→1       55    0.007%
2→2    14796    1.995%
2→3      674    0.091%
3→0     1754    0.238%
3→1      124    0.017%
3→2      534    0.072%
3→3     4427    0.597%

tot.  741655
```

Figure A.2. Sample display for character-by-character analysis program.

total number of transitions: This is the number of requests for memory transitions.

# of M0→M1→M0 transitions: This is the number of occurences of a M0→M1→M0 transition.

The next 5 lines give the character usage among the four memories. Both the total number of characters and the percentage of the total visible characters for each memory is given.

The frequency of occurence for each of the special codes (shift, access, etc.) is then listed along with the percentage relative to the number of internal PLATO characters.

At the top right:

bits/character: This is 6 bits times the number of visible characters divided by the number of formatted characters. The number in parenthesis includes the data/control bit.

This number plus 12% fill is given in the next line, in the same format.

The limit by Shannons' bound is calculated from the character distribution using the formula given in 5.1.

The remainder of the display gives the transition information. For example, the entry labeled 0→0 indicates that out of 741655 visible characters, 619263 were displayed from M0 without any transition. Therefore, 83.50% of the time, the base memory was M0, and the next character was also in M0. Also, summing the four entries which indicate that the final memory was M0 gives the total number of characters displayed from that memory, which matches the entry for M0 on the left side. This provides an internal consistency check.

This program was also used to provide the information for the character frequency graphs drawn in Figures 4.3 and 4.4. A variation of this program was used to determine which type of display information was predominant. Another variation was used to try to find what length lines are common; however, it was decided that the sampling technique destroys that information. If the sampling program were modified, analysis of lines would be possible.

Future uses of the character-by-character analysis are: studying the internal format with regards to transmitting internal codes directly to the terminal, and analyzing the effectiveness of any system change.

A.3 Word-by-word Analysis Program

This program provides the word frequency distribution information for Chapter 6 from the data generated by the sampling program. First, the text is scanned for delimiters, which are all non-alphabetic characters. Anything between delimiters is considered a word. The words are kept in a table in ECS, in alphabetical order, which is updated to a disk file periodically. Each time a work is found, a binary chop is used to find the word in the table. If it is not there, it is inserted in the proper position. Each table entry is two 60 bit words long. Up to 17 6 bit codes are stored per entry. The remaining bits are used for frequency information.

While collecting words, the table is allowed to grow to 6601 entries. Then it is sorted by frequency and the amount representing 3/4 of the total words are retained. This is usually around 600 entries.

The table is then resorted alphabetically, and the processing continued. A typical sample represents approximately 100,000 words.

The following calculations are performed on the table: sort by frequency, percentage of total words for each word, percentage of total characters for each word, percent savings for each word, and a running total for each of these.

The most cumbersome part of this program is the enormous amount of time needed to create the original word frequency table. Running under low system load, this takes several hours real time, not necessarily consecutively. The table lookup is expensive because the entire table will not fit in central memory. The binary chop was selected because it is a fast search routine, and it could be performed without transferring the entire table into central memory. Future uses of this program would be to study character grouping different than words, such as dipthongs. However, to be truly useful, the word gathering part must be made faster. Writing it in Fortran would be one possibility.

A.4 Word-by-word Analysis of Source Files

As a preliminary study, a program written in Fortran was used to compile word frequencies from lesson source code. However, it was felt that this could not be representative as it did not include the effect of repeated displays. Also, it required guessing the lesson mix to simulate the system load. However, for specific areas, such as one group of students, a reasonable approximation of the word frequency order can be gotten by scanning the lessons that are included in their curriculum.

APPENDIX B

AN EVALUATION OF CERTAIN VOICE SIGNAL CHARACTERIZATION TECHNIQUES FOR A LOW BANDWIDTH SPEECH RECOGNITION SYSTEM

James Lee Oppenheimer

TABLE OF CONTENTS

1. INTRODUCTION

As computer based systems play an ever more integral role in our society, the need for more natural and efficient man-machine interaction has become increasingly apparent. The current means of interaction (such as keysets and light pens) often require special skills or are slow and cumbersome to use. The development of an input system involving previously acquired skills and more reflexive, familiar, and simple mechanisms could considerably enhance the usability of computers.

The use of speech is one such approach to attaining this goal. Speech is one of the most universal, earliest learned and most effective forms of human communication. In addition, the special applications of speech input are numerous. In an educational environment, speech can play a vital role in teaching foreign languages and reading (since nonreaders can't type responses). It can provide a means for the handicapped to use computer systems and can free users to use their hands for related tasks.

One approach to handling a high number of interactive terminals on a system is to run a low bandwidth output line from the terminal to a central processor. An alternate method is to multiplex one high bandwidth line attached to a number of terminals, the rationale being that keypresses occur at a relatively slow rate. This form of system architecture is particularly suitable for situations in which a number of interactive terminals are located in close physical proximity, such as airports, business offices and educational environments.

Developing speech recognition systems for this type of configuation presents a number of serious problems, since highly complex speech information has to be transmitted with a relatively small number of bits if response

times are to be reasonable. To achieve this low bandwidth, the speech information has to be compacted into as reduced a form as possible by properly extracting and encoding the key aspects of the speech signal, and by taking advantage of the inherent redundancy of speech. Such a system was developed for the PLATO computer aided instruction system[1,8].

The design goals for such a system were that it recognize isolated words reliably from multiple speakers if possible, that it be fairly inexpensive, and that it be compatible with current PLATO architecture.

The object of the research described in this thesis was to evaluate the performance of the pre-existing speech recognition system[2], with an eye towards its possible use as an educational tool, and to then improve that system primarily through hardware modification. The specific improvements sought were an increase in the reliability of recognition, and a decrease in the number of words of information generated to describe an utterance.

Chapter 2 of this thesis describes the original recognition system and the rationale behind that particular approach. Chapter 3 deals with the system used to evaluate performance and the initial baseline results obtained. Chapter 4 describes the various modifications that were implemented and their effect on performance. The final chapter provides a summary of my conclusions and gives suggestions for further research.

2. BASIC SPEECH RECOGNITION SYSTEM

2.1 DESIGN APPROACH

The design approach of the PLATO speech recognition system (designed by Jim Parry) was oriented toward the need to eliminate a large amount of the nonessential information contained in an utterance while retaining a reasonable level of recognition. In addition, it was necessary to send information in a form compatible with the 10 bit input word format for PLATO. The guaranteed key input rate from a PLATO terminal, based on the polling rate of a site controller, is 2.5 keys, or 20 information bits, per second. Larger rates are possible, though, if site controller usage is less than maximum. To accurately represent a speech signal, bandlimited to 3 kHz, using sixteen levels of quantization, requires 24,000 bits per second of input, based on the sampling theorm. Hence there is a need for reducing the information sent down the line.

Previous research[3,4] has indicated that a combination of zero crossing and energy measurements can provide a reasonably good characterization of an utterance. Zero crossing measurements are particularly well suited since they are easily produced digitally, are virtually independent of speaker volume, and are less sensitive to speaker variation than spectral measurements.

Two different zero crossing measurement are made. The first is a raw zero crossing rate (between 400 Hz and 5.6 kHz). This measurement is valuable for distinguishing phoneme types (such as vowels and fricatives). The second zero crossing measurement is taken after the signal has been bandpass limited to the range of 1 to 3 kHz. This band, which is known as the second formant, plays a vital role in the distinguishing of vowel sounds.

The energy measurements provide important additional information. High peaks tend to indicate voiced or accented phonemes. In addition, the energy magnitude is valuable for determining the beginning and end points of an utterance, and indicating the positions of syllables.

2.2 SPEECH INPUT HARDWARE

The actual audio input hardware (Figure 2.1 and Appendix 1) was designed to be attached to the external input and output jacks of a PLATO IV terminal. The speech signal is transduced by a microphone which then drives an amplifier that has its gain controlled by the central computer. The three measurements are performed in parallel. The energy measurement is taken by full-wave rectifying the speech signal, integrating it over a 10 ms. period, and converting the result into digital form with a tracking A/D converter. The two zero crossing rates are determined by amplifying the signal to the point of clipping, putting the result through a Schmitt trigger and rate multiplier, and then counting the number of pulses in a given time with a counter. To prevent background noise from being transmitted as speech information, these two zero crossing measurements are inhibited whenever the signal intensity falls below a certain level (the 0001 point of the A/D converter).

Once these three signal measurements are made, they are sequentially sampled at a rate of 100 samples/second each. Each sample is compared to the value of the previous measurement of that type that was sent to the computer, which is stored in a register. The new value is sent only if it differs from the old value by a threshold which has been set by the computer.

If the decision is made to send a new value, the four bits of the measurement, two bits to indicate measurement type and two bits that give the approximate logarithm of the time elapsed since the previous key was

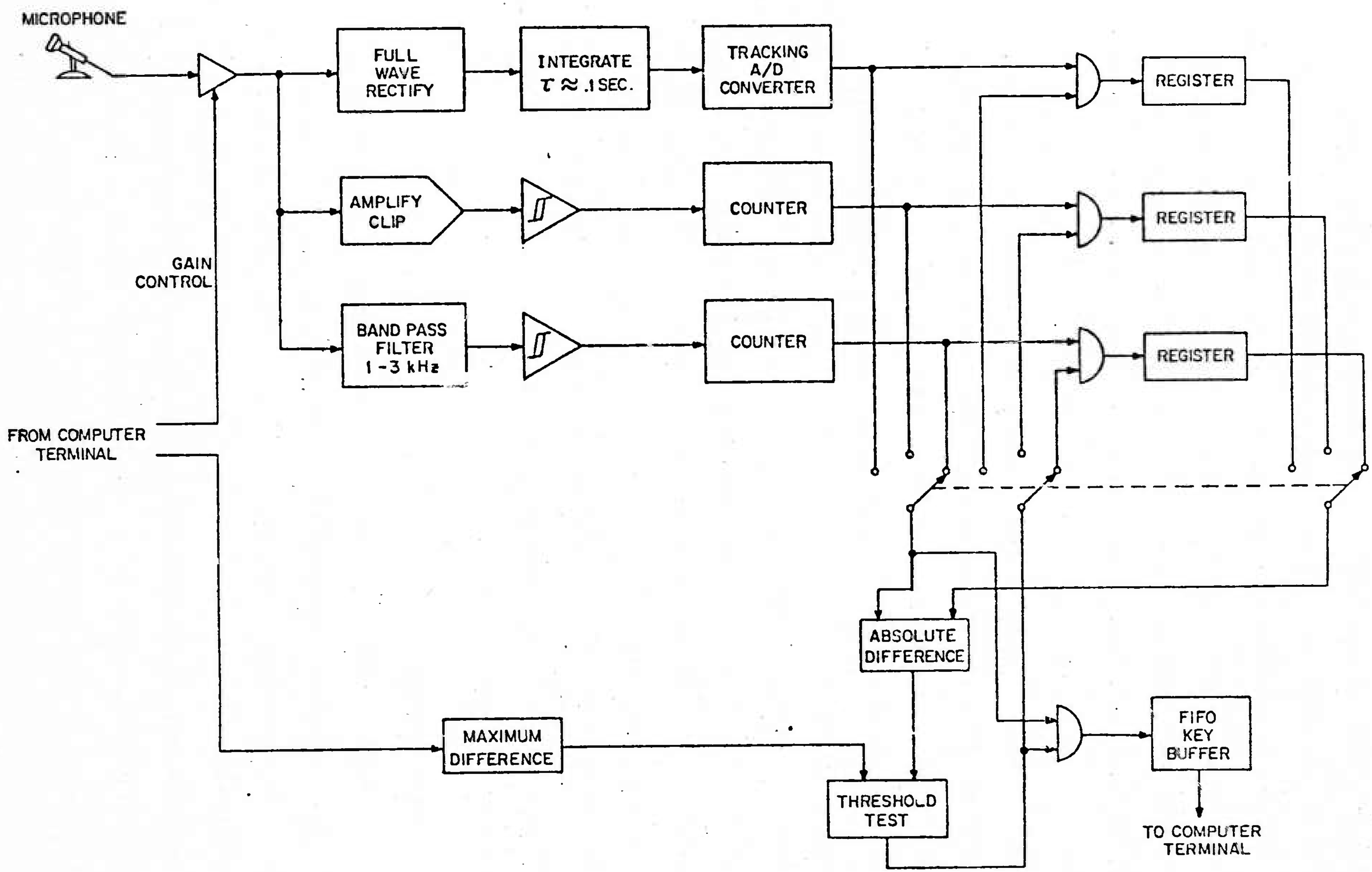


Figure 2.1 Block Diagram of Speech Input Device.

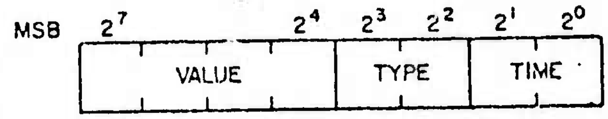
generated, are formed into a "key" and placed into a 64 word FIFO buffer (Figure 2.2). The keys that are stored in the buffer are then clocked out to the computer at the slow rate of either 6 or 12 keys per second, depending on the particular device. With this system, if the value of a parameter remains essentially unchanged over a period of time, then only one word of information is sent, significantly reducing the number of keys that need to be sent.

Once a sequence of keys has been received by the central computer, the speech input software performs smoothing of the sampled values and constructs a series of three time-normalized waveforms, each consisting of 20 data points equally spaced in time. During this process, the energy waveform is normalized to the highest level amplitude sample. In addition, the total duration of the utterance is determined by summing the timing values received.

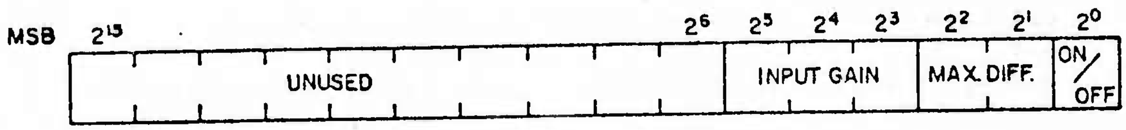
In order to be able to perform recognition, the system must first be trained with at least one utterance of each of the words in the vocabulary. During training the 60 four bit samples (20 for each measurement type) are stored in 6 words of PLATO common storage, with a seventh word being used to store the duration of the utterance (plus other data) and an eighth word to store a character string for identification.

2.3 RECOGNITION SOFTWARE

To perform the actual recognition task, the measurements taken on the utterance to be recognized are compared with those of the stored vocabulary. The duration is compared by taking the difference in the two values over the square of the sum of the values. Comparing in this way tends to place a greater emphasis on small duration differences in short utterances than on small differences in longer ones. The three measurement waveforms are compared by computing a hyperbolized area between the stored and uttered



INPUT WORD FORMAT



OUTPUT WORD FORMAT

Figure 2.2 Input and Output Word Formats for Speech Input System.

waveforms. These four "scores" are then weighted according to their relative importance and summed to form a total score that reflects the similarity between the utterance and the vocabulary entries. The eligible vocabulary word with the lowest score is then taken to be the correct word. If desired, certain words can be put into separate vocabularies or can be tagged as eligible or ineligible depending on what words are likely to be spoken. In this way processing time is reduced and recognition improved.

As word matching takes place, the new utterances can be averaged with the initial training utterances to refine the initial data or to adapt the vocabulary to a particular speaker. The weight given to the new utterance is dependent on the number of previous averagings of that word according to the formula

$$\text{weight} = .9 - 1/(1.5 + \text{number of previous averagings})$$

$$\text{new value} = (1 - \text{weight}) \times \text{old value} + \text{weight} \times \text{new value}$$

As an illustration, the waveforms generated by the words "history" and "hello" are shown in Figure 2.3. For these examples, one can quite clearly break up the words into the sound groups that make up the utterances. It should also be apparent that it is easy to differentiate the two words by comparing waveforms.

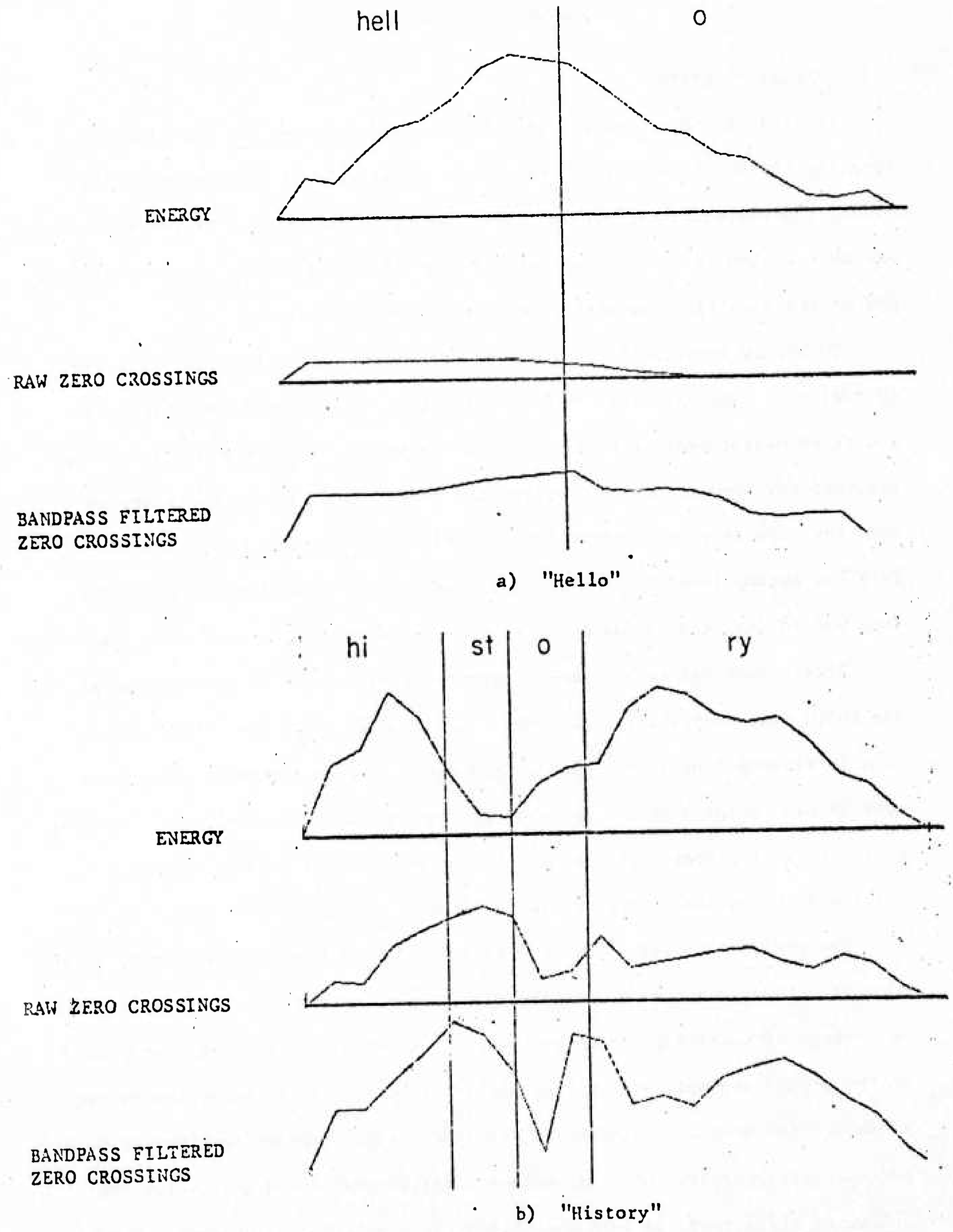


Figure 2.3 Waveform Produced by Utterances of a) "Hello" and b)"History".

3. PERFORMANCE EVALUATION

3.1 EVALUATION SYSTEM

In order to determine the usefulness of this speech input device, identify its shortcomings, and ascertain the effects of hardware modifications, a performance evaluation system had to be developed. It was necessary for this system to use an input signal that reflected actual usage conditions and at the same time was easily repeatable from test to test.

To insure repeatability, a set of utterances was prerecorded on a reel to reel tape recorder directly from the output of the amplifier that feeds the three measurement sections of the voice input. Since the amount of time required for word processing varies with system load, it was necessary to have the tape recorder started and stopped by the recognition software. This was accomplished by connecting a relay to the tape recorder power line that was in turn controlled by a bit in an external output word from PLATO.

Precautions had to be taken to prevent keys from being lost either at the PLATO site controller (since the guaranteed 2.5 keys per second input rate is exceeded) or at the central processor (due to automatic interrupts made likely by the high CPU usage of the recognition software). To insure that keys were properly received, a handshaking scheme between the voice input and the central computer was instituted. (See Appendix 2.)

The evaluation system was used to accumulate a number of different types of data. First a set of overall average statistics was stored, including the percentage of correctly recognized words, the percentage of words recognized on the second attempt, the average number of keys sent per word, the average score of the recognized utterances, and the average processing time per word per recognition pass. Next, a confusion matrix was stored indicating the number of times that one word was mistakenly identified as another. Finally,

an average of the score separation between a properly recognized word and the next most likely choice was computed for each properly recognized word. This measurement characterizes the effectiveness with which the scoring system differentiates properly recognized words from the others, and provides an indication of the tolarance of the system to possible variations in the pronounciations of an utterance.

All data accumulated was later analyzed by a statistics package that computes the cross correlations between the various measurement types, and the point biserial correlation between word prediction of a particular measurement type and correctness of response. The mean and standard deviation of each measurement type was also calculated.

The primary test used to judge performance was a recording of the digits (zero through nine), each spoken 6 times by 3 different speakers. A man with speech training was used as the first speaker and for the training utterances. A 30 year old woman and an 11 year old girl, both recorded at different volume levels, were the other two speakers. This test was chosen for a number of reasons. First, it has been the standard test for most previous speech research, secondly PLATO or any CAI system makes a critical use of numbers for responses and for choosing branching alternatives, and thirdly the digits are quite difficult to recognize as they are almost all one syllable words. After training is performed, the new utterances are averaged in with the training utterances as would probably be done in any system implemented. Unless otherwise indicated, all tests were performed using a gain of 20 and all difference thresholds set to one.

3.2 BASELINE PERFORMANCE MEASUREMENTS

Before any modifications were made a set of baseline performance statistics was accumulated as a means for evaluating changes resulting

from these modifications. These tests were run under the conditions mentioned in the above paragraph. Two sets of tests were run. The first set consisted of single speakers, with that speaker providing the training utterances. The second involved all three speakers, with the first being used to train the system. All tests were run at least twice and all results indicated represent the average of those tests.

For the male speech trained subject the recognition rate averaged 80% correct with an average of 27 keys being sent per utterance. The keys sent were approximately equally divided among the three measurement types, although there were slightly more energy keys (43%) sent than zero crossing keys.

For the female speaker (recorded at a somewhat higher volume level) the recognition rate was 90% with an average of 34 keys sent per utterance. The difference in the number of keys transmitted was due primarily to a doubling in the number of bandpass filtered keys sent.

The recognition rate for the child (recorded at a somewhat lower level than the male speaker) was 72% with an average of 22 keys sent per utterance.

The multiple speaker test used the man's, woman's, and child's utterances in that order, with the male voice being used to train the system. The vocabulary stored was modified throughout but the weighting factor was initialized to 1/2 at the beginning of each speaker. The recognition rates obtained were 80%, 66%, and 60%, respectively, for an overall response of 68%, which is considerably lower than for the individually run tests. The number of keys transmitted per utterance was 29, roughly corresponding to the average of the individually run tests, as would be expected.

The average recognition processing time per vocabulary entry was approximately 11.5 ms. The score separations showed considerable variations

from test to test but gave a good indication of which words were the most different from the others. (Figure 3.1) The words "three" and "eight" showed the largest score separation, a somewhat unexpected result since eight was frequently misrecognized, although much of the time this was due to the system being unable to match any word with the utterance. The score separations for the multiple speaker tests were surprisingly about the same as for the individual tests.

Only between 30 and 50 percent of words improperly recognized were correctly chosen on the second attempt, depending on the test. This indicates that confusion resulted from more than just a single word being similar to the spoken word, otherwise recognition on the second pass would have been close to 100%.

While the actual words that were incorrectly recognized varied from test to test, certain words were consistantly confused. (Figure 3.2) One and five were frequently mistaken for each other as were one and four, and four and five. These errors are a bit difficult to explain since the "ai" and "n" phonemes have rather dissimilar frequency spectrums and hence different zero crossing rates. A significant amount of noise was observed on the test tapes which could have caused erroneously high zero crossing readings when low volume signals are present, thus causing the low energy "n" sound to appear similar to an "ai" sound.

The word eight caused the most difficulty in recognition as might be anticipated since it has low energy initial and final phonemes. The words seven and three proved the easiest to understand. This can be attributed to the fact that seven is the only multiple syllable word used and three has a strong voiced phoneme at the end.

WORD	SCORE SEPARATION
one	111
two	314
three	491
four	175
five	77
six	416
seven	148
eight	367
nine	223
zero	321
AVERAGE	265

Figure 3.1 Score Separations for Multiple Speaker Tests (Out of a Possible Separation of 1023).

Word Matched Incorrectly

Word Spoken	one	two	three	four	five	six	seven	eight	nine	zero	NO MATCH
one	0	1	0	3	5	0	2	0	0	0	0
two	0	0	0	0	0	0	0	1	0	1	1
three	0	0	0	0	0	0	0	0	2	0	1
four	3	0	0	0	0	0	1	0	1	0	0
five	2	0	0	4	0	0	0	0	1	0	0
six	0	1	0	0	0	0	2	0	0	0	0
seven	0	1	0	0	2	0	0	0	0	0	0
eight	0	0	2	0	1	3	1	0	0	1	1
nine	1	0	1	0	0	0	3	1	0	0	0
zero	0	2	0	2	0	0	0	1	0	0	1

Figure 3.2 Confusion Matrix for Multiple Speaker Test.

The most important result indicated by the statistical package was the relative importances of the various measurement types. (Figure 3.3) The bandpass filtered zero crossing rate showed the highest correlation between predicted response and correct response. Next came the raw zero crossing rate with a slightly lower correlation. The duration and energy measurements had considerably lower correlations than the bandpass measurements. All the correlations for the multiple speaker tests were lower than the individual tests.

A series of tests were run to determine the sensitivity of the system to the gain of the input signal, with an eye towards the development of an adaptive volume control system. The adaptive volume control software would monitor the largest energy transmitted over a number of utterances and would increase gain if it were consistently below a certain value and, conversely, decrease gain if it were consistently above it.

Testing was somewhat difficult at either extreme of the possible amplifier gains. At low gain some of the recorded utterances did not have sufficient amplitude to cause the noise threshold to be exceeded, while at high amplitudes background noise was high enough to exceed the threshold. As a result, only gains of 15 through 30 were tested.

Both recognition and number of keys generated seemed to be effected by gain. (Figures 3.4 and 3.5) The effect on recognition was not particularly dramatic, with only a 5% spread over the range of gains tested. The effect on the number of keys was much more substantial, with a 13 key difference registered for only a 2 fold change in gain. The major variation was in the raw zero crossing measurement. The number of raw zero crossing keys dropped from 770 to 290 while the number of energy keys varied by less than 100. One possible explanation for this is that with higher gain the noise

Correlation between scores of	Single Speaker	Multiple Speaker
duration and energy	.469	.818
duration and raw z-xings	.139	.059
duration and filtered z-xings	.066	.117
duration and correct response	.095	.041
energy and raw z-xings	.092	.063
energy and filtered z-xings	.080	.162
energy and correct response	.121	.065
raw z-xings and filtered z-ings	.414	.063
raw z-xings and correct response	.238	.193
filtered z-xing and correct response	.304	.284

Figure 3.3 Correlations Between Measurement Scores.

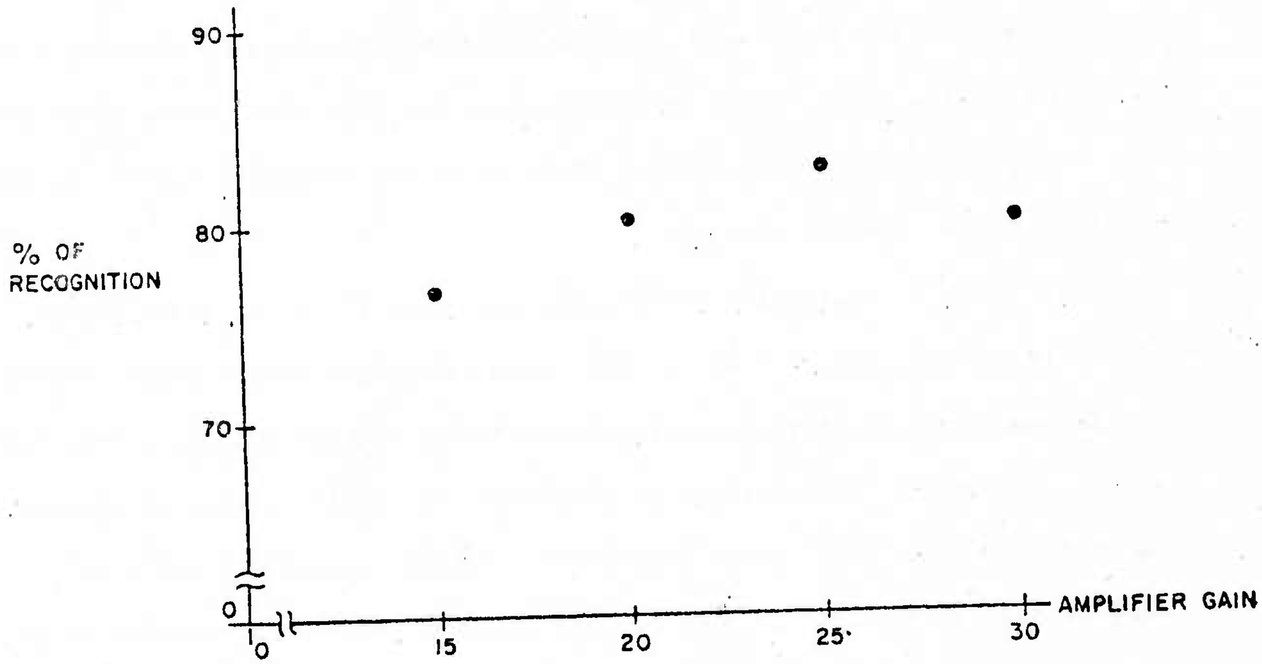


Figure 3.4 Graph of Recognition vs Amplifier Gain.

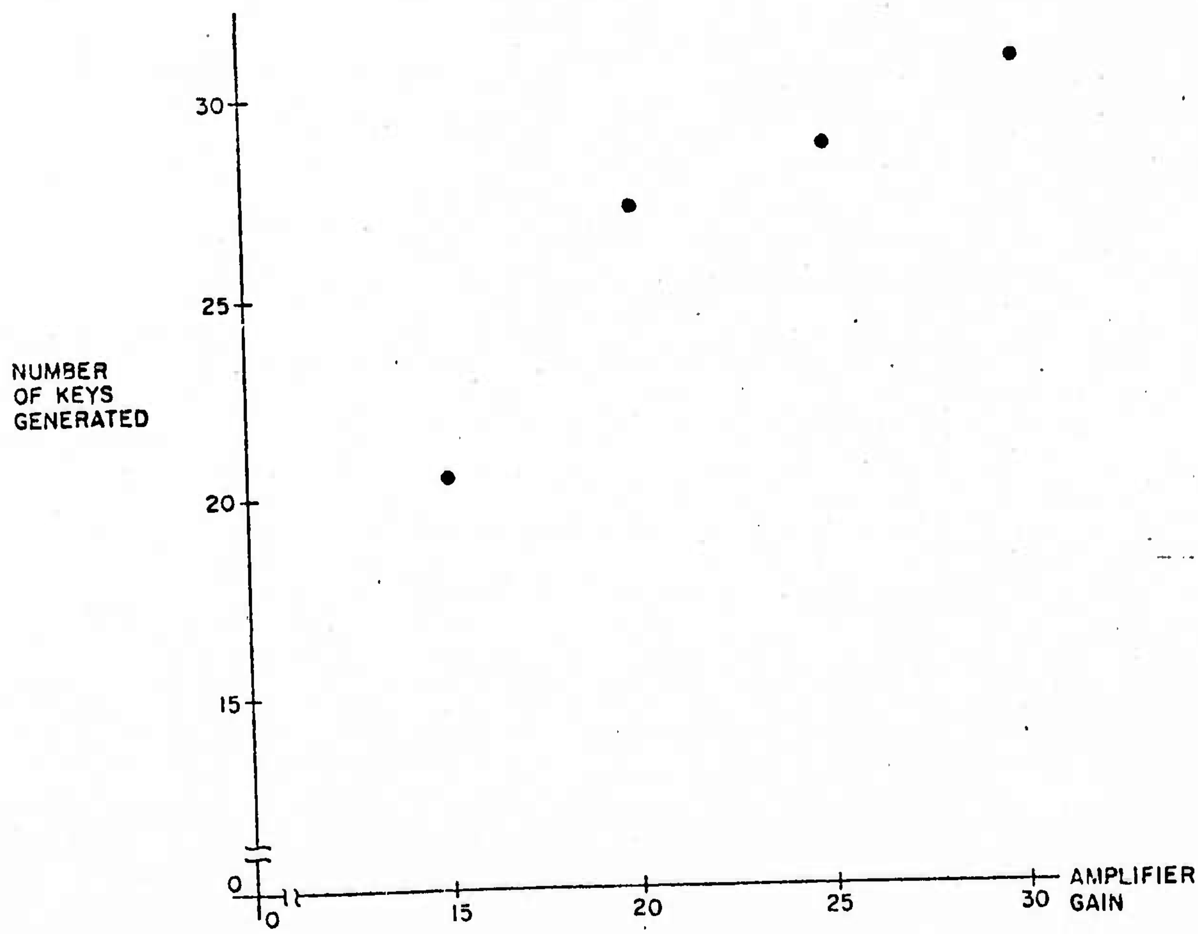


Figure 3.5 Graph of Number of Keys Generated vs Amplifier Gain.

threshold is exceeded at an earlier point in an utterance, thus causing more zero crossing data to be sent. Another reason for this may be that more noise is being amplified beyond the threshold level of the Schmitt triggers in the zero crossing measurement sections.

These tests indicate that while gain does play a role in determining system performance, the effects are not really dramatic at the gains tested. It would seem that consistency in amplitude would be more important than the absolute amplitude. Therefore an adaptive system would probably do more harm than good, since it would cause more errors through changes in amplitude (hence changes in the energy curves) than might be avoided by running at an optimal amplitude.

4. HARDWARE MODIFICATIONS

4.1 INTRODUCTION

As mentioned, a series of hardware modifications were made to the voice input system in an attempt to improve the reliability of recognition and, at the same time, reduce the number of keys generated per utterance. These modifications were implemented in such a way as to allow them to be easily switched in and out of the circuit. In that way, the effect of any modification could be isolated and tested separately. To test the modifications, the individual male speaker test and the multiple speaker test were used. The individual modifications are each described in separate subsections of this chapter.

4.2 MULTIPLE DIFFERENCE THRESHOLDS

The original system utilized one difference threshold for all measurements. Since the different measurements vary in importance it was speculated that using an individual level for each type might provide more optimal performance.

Such a system was implemented by using 6 bits of the PLATO external output word to send three threshold levels to the voice input device. In the original device a single two bit difference threshold was used as one input to a four bit comparator, with the other input being the difference between the current and previously sent value of any measurement. The old system has been modified so that the three threshold values (which are stored in latches) are inputed to the comparator through a multiplexer that is controlled by the two bit clock that indicates the measurement type.

To reduce the amount of testing necessary it was assumed that the effect on recognition of changing one threshold value was independent of the other

values. In that way one value could be altered while the other two were held at some constant value. This is not a totally valid assumption for a couple of reasons. First, the measurements themselves are not orthogonal as is demonstrated by the high cross correlations between various measurement types. The two zero crossing measurements, for example, have a .4 cross correlation (during the single speaker). Secondly, since the prediction made by the system is the result of the addition of four weighted factors, the other three factors may mask the effects of the measurement being tested. Nevertheless, testing all possible combinations of thresholds would require 128 tests, a prohibitive number in terms of testing time required.

The varying of threshold levels turned out to have a major effect on both recognition and the number of keys generated, as illustrated by Figure 4.1 and 4.2. For all three measurement types the number of keys generated drops in a somewhat exponential manner as the threshold increases. The recognition rate also drops with the number of keys in a somewhat exponential way for the two zero crossings measurements. For the energy measurement, on the other hand, recognition varies only slightly with the number of keys sent.

A good approximation to the optimal threshold level is the point where the rate of decrease in recognition becomes greater than the rate of decrease in keys sent. At this point the benefit in terms of increased recognition becomes greater than the price paid in terms of keys generated.

Examining the data reveals that by this criterion setting all the threshold levels to one (meaning a change greater than one is necessary) as in the original system, provides the optimal performance. Nevertheless, 93% recognition was attainable using a bandpassed zero crossing threshold of zero with the other thresholds set to one. While this is not optimal in

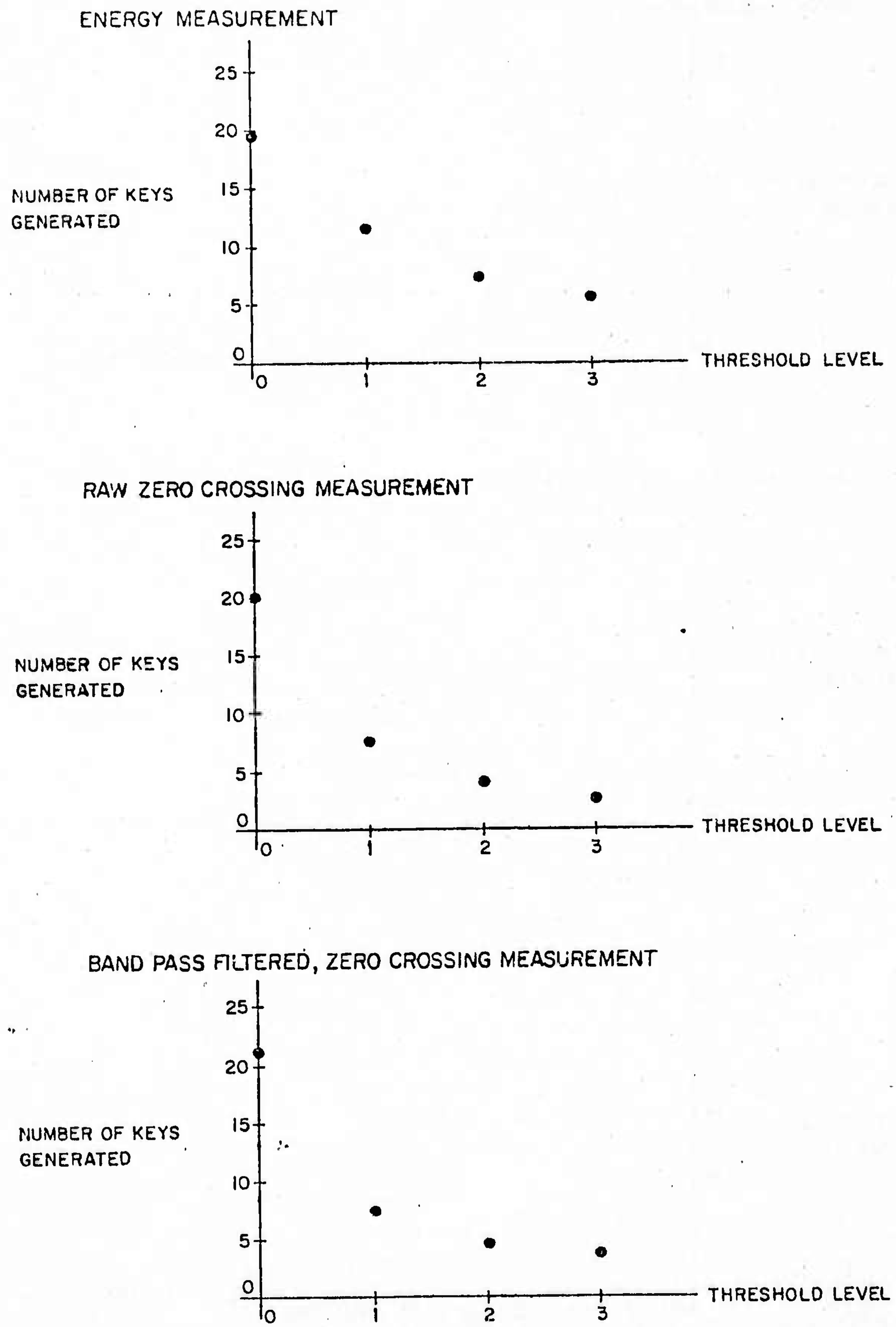
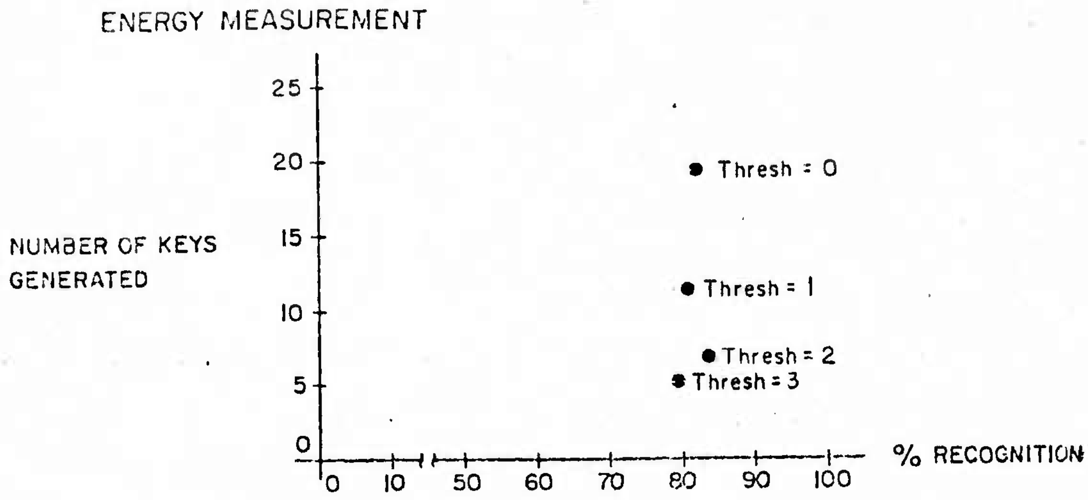
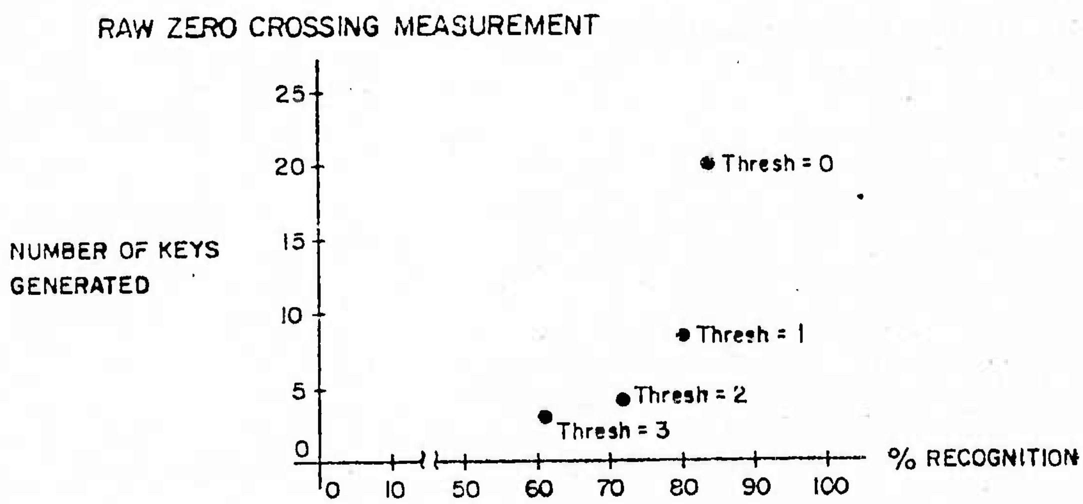


Figure 4.1 Graphs of Recognition Reliability vs Different Threshold.





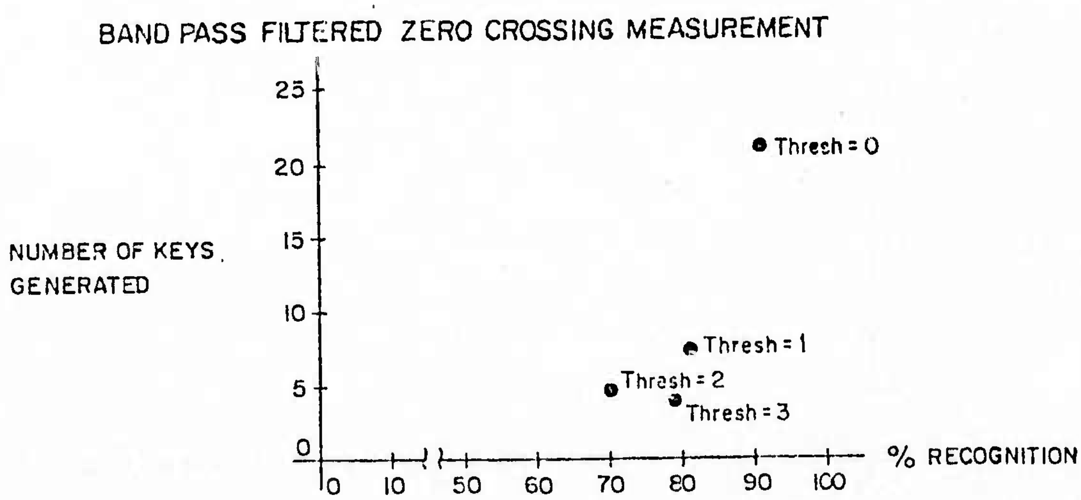


Figure 4.2 Graphs of Recognition Reliability vs Number of Keys Generated for Various Difference Thresholds.

terms of number of keys vs recognition obtained, it is a substantial improvement in recognition, achieved at a much lower key transmission cost than would have been possible if all thresholds had to be set to zero. Since changes in the energy threshold have little effect on recognition but significant effect on the number of keys generated, it would seem beneficial to set this threshold to a level greater than one. It is also possible that for certain vocabularies, some measurement types provide more critical information than others. The vocabulary consisting of yes and no, where the zero crossing rates provide the most important data, is an example of this. If this is the case then it would be beneficial to set that threshold to a low level and the others to higher ones. The drawback of such a system is the lack of any clear guide (besides extensive testing) to choosing levels for a given vocabulary.

4.3 TWO PASS SYSTEM

While the original scheme for generating keys is intended to eliminate all but the most vital characteristics of an utterance, it is still possible for short transient sounds that are not critical for recognition to cause additional keys to be sent. Also short bursts or spikes of noise can cause the transmission of spurious information.

A possible remedy to these problems is to require that the difference threshold be exceeded on two successive sampling passes in order for a change in a measurement to be considered legitimate. In this way, a change that takes place within only one sampling interval (as is likely to be the case for a noise burst) will not cause extra information to be transmitted. Another solution that is easier to implement is to double the length of the sampling interval, thus integrating each of the measurements over a longer period of time.

The circuit implementation of the two pass key generation scheme was inserted between the measurement comparison and the FIFO control circuitry. A set of three memory elements (one for each measurement type) is used to indicate whether a threshold has been exceeded on the previous sampling pass. The register's outputs are used to gate clock pulses that determine whether data is inserted into the FIFO. (See Appendix 3.)

The results of the two pass modification were a decrease in the number of keys generated (as would be expected) and a degradation in recognition ability. The number of keys generated was reduced by a factor of one third (from 26 to 17 keys). All three measurement types were reduced by approximately the same percentage, with the energy measurement reduced by a slightly higher degree.

Recognition became somewhat more erratic with a spread of 9% in results. The recognition rate for the single male speaker was 70%. The result of the three speaker test was 73.33, 61.67, and 45% for man, woman, and child, respectively. The correlations also were reduced, each by almost equal factors, with the time correlation being reduced by the highest percentage.

The results of converting to a slower clock were similar to those of the two pass system. The number of keys went down but by a somewhat higher degree. The average number of keys transmitted was 15. The recognition rates also became more erratic, with a considerable variation from test to test. The reliability of recognition varied from 68% to 80%.

4.4 FINISH KEY

There are two drawbacks of the old voice input device that can be remedied with the addition of a finish key at the end of an utterance. The first is that any sound received by the input between the completion of an

utterance and the emptying of the FIFO buffer will generate keys that will be included as part of the utterance. A student using the system, therefore, would have to remain silent until all keys were transmitted (a period that could amount to a few seconds).

A finish key (with type bits equal to 11) inserted after the occurrence of a predetermined period of silence beyond when the input signal falls below the noise threshold, will act to indicate the end of an utterance. Any keys received after the receipt of the finish key would then be ignored, or if multiple utterances are desired it can act to demarcate the two utterances.

A second problem is that the utterance duration calculated is quite inaccurate since it is based solely on the two timing bits of a key. Since 20 ms or greater is the highest timing gap between keys that can be indicated, longer gaps between keys will cause considerable error in duration.

The finish key can be used to solve this problem by providing an accurate duration measurement in the six remaining bits. If one assumes that words can never by more than 1.3 seconds long, then duration can be indicated to within 20 ms accuracy.

The use of a more accurate duration measurement lead to a noticeable improvement in recognition ability. For the single speaker test recognition was increased from 81% to 86%. For the multiple speaker test the results were 88%, 71.6%, and 66.6% for an average rate of 76.5%, a 10% increase in correct responses. More significantly, the correlation between the response predicted by the time measurement and the actual correct response more than doubled for the individual test and nearly tripled for the multiple speaker test.

Data accumulated during the testing of the finish key shows that the old method for calculating utterance duration tended to produce durations

that were greater than those obtained by the more accurate measuring system. This is understandable since an artifact of the duration calculation software is that all durations get calculated as a value higher than that transmitted (i.e., the 20 ms measurement separation gets treated as 60 ms). The average error per utterance was 41 ms, with the error resulting from overestimates about 75% of the time.

It appears therefore that the incorporation of a finish key leads to a significant (though not dramatic) improvement in recognition reliability without causing a substantial increase in the number of keys generated.

4.5 DETERMINING WORD BEGINNING AND END POINTS

One of the problems faced by most speech recognition systems is that of determining the beginning and end points of words. Phonemes that have low signal energy (such as fricatives, nasals, liquids, and unvoiced phonemes), when located at the beginning or end of a word, can make this problem particularly difficult because they often are hard to distinguish from background noise[5].

This endpoint problem was particularly acute in the system described here because the noise threshold used to determine whether speech information is present is quite high (being the smallest resolved value of the A/D converter, or 1/16 of the maximum measurable excursion of the signal amplitude). In addition, to avoid sending unnecessary keys to PLATO, determination of the endpoints had to be made by the input hardware.

In an attempt to more accurately determine an utterance's endpoint, a set of two thresholds more sensitive to events at low amplitudes was utilized to predict the start of an utterance or indicate that speech information is still present at the end of an utterance. The thresholds used were a low

level amplitude threshold and a high level zero crossing threshold. The amplitude threshold was set at .05 volts (one quarter of the old threshold) and the zero crossing threshold at 2.8 kHz (higher than the zero crossing rate observed in a typical terminal environment).

If either of these two thresholds is exceeded, that indicates the likelihood of speech information being present, so keys are then placed in the FIFO buffer with the data ready flag being held low, preventing the transmission of the data to PLATO. Only when the original high level threshold is exceeded is the data that has been collected transmitted. Should the signal levels drop below both thresholds before the high level threshold is exceeded, the information stored is cleared, since no word was really present. Since most phonemes last 50 ms or less, if 80 ms elapses after the low level threshold has been exceeded without the high level threshold having been exceeded, then the data is also cleared on the assumption that the signal measured was noise. The same process takes place at the end of a word except that keys are inhibited from entering the FIFO rather than the buffer being cleared, when valid isn't present.

Contrary to expectations, the endpoint prediction circuitry had very little substantial effect on performance. The number of keys generated did not increase at all above the rate of 25 keys per utterance obtained in the baseline testing. The rate of recognition also did not change drastically with a 78% correct result (slightly but not significantly lower than the baseline tests). Looking at the raw key output revealed that even with strong fricative sounds, there were no long strings of zero crossings keys at either the beginning or ends of utterances, unless very strong emphasis was placed on these sounds.

A possible explanation for the ineffectiveness of this modification could be the noise limiting circuity of the Plantronics microphone, which

is designed with an amplitude threshold to eliminate all low amplitude signals. Additionally, pauses between the uttering of a fricative and the next phoneme could cause the keys generated by the initial phoneme to be lost. A short pause between the point where both thresholds go low and the FIFO is cleared might eliminate this problem.

4.6 LOGARITHMIC COMPRESSION

It is often the case for speech signals that small amplitude changes at low volumes have a much greater significance than larger changes at high volumes. There is a 200 fold intensity difference between the softest and loudest consonant. The intensity spread between vowels on the other hand is only three to one, with the weakest vowel being equivalent in intensity to the strongest consonant[6]. Since there are more consonant phonemes than vowel phonemes, there is a need for greater sensitivity at low amplitudes. Logarithmically compressing the speech signal leads to this low amplitude sensitivity. An additional justification for using logarithmic compression is that the ear perceives intensity in a relatively logarithmic manner.

The logarithmic compression is performed at the output of the integrator circuit since this is the point where a DC signal representing the peak amplitude of the signal exists. The amplifier was built using an operational amplifier with a diode in the feedback loop. It was necessary to be sure that the input of the log amplifier was biased slightly positive when no input was applied and that the output was biased negative somewhat to prevent premature triggering above the noise threshold.

Figures 4.3 and 4.4 illustrate the difference between the waveforms generated by the linear and logarithmic amplifiers. The logarithmically compressed waveform contains very few low level values. The measurements

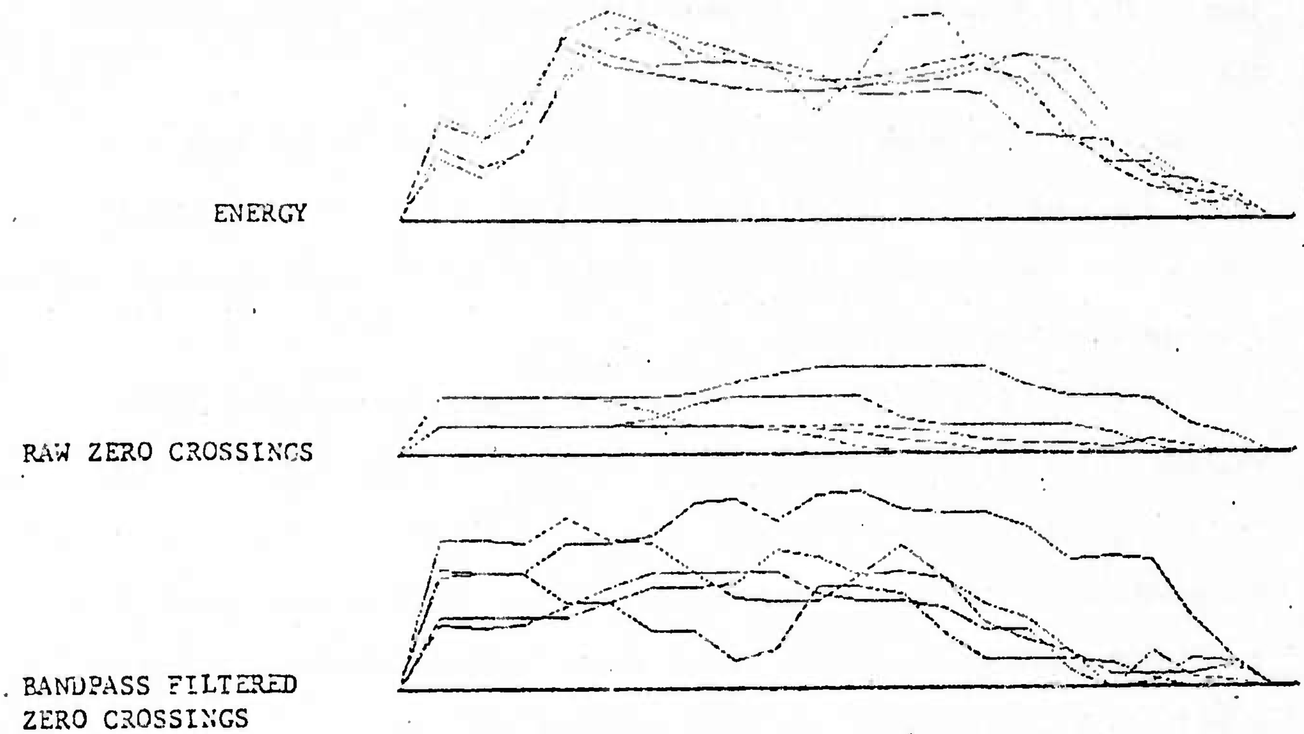


Figure 4.3 Waveforms Produced by Linear Amplification.

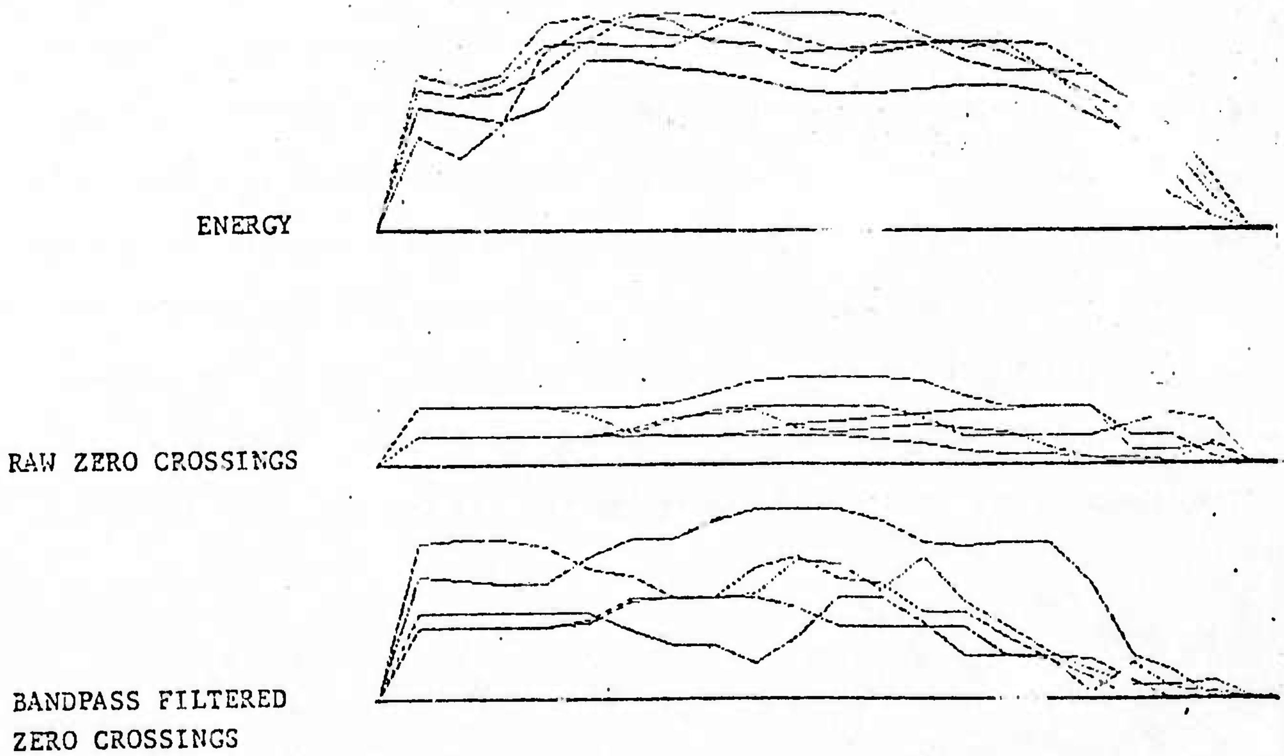


Figure 4.4 Waveforms Produced by Logarithmic Amplification.

jump to the high portion of the range almost immediately. As a consequence the recognition reliability decreased to some degree.

Recognition rate went down to 73% for the individually run male speaker test. The number of keys generated decreased to 21 keys per utterance with the decrease coming from a drop in the amplitude keys as would be anticipated from the waveforms observed.

The sudden rise in the energy measurement indicates that the signal reaches a high amplitude quickly after passing through the noise threshold. The logarithmic amplifier tends to emphasize this rapid rise since the small excursions at low amplitude cause greater changes in measurement value than for linear amplification. Hence it is likely that the signal is rising so quickly that the low levels are never sampled.

The sudden rise in signal level could be due to the noise limiting circuitry of the Plantronics microphone. Also, since there is a considerable range of phoneme intensities as previously mentioned, it is possible that the low energy phonemes that fall below the voice input device's noise threshold are followed by high level vowel sounds. The likelihood that the amplitude of the low energy phonemes falls below the noise threshold is increased by using silicon diodes to rectify the speech signal, preventing voltages below .7 volts from passing through. A possible remedy to this problem would be to use germanium diodes (with their .3 voltage drop). Use of these diodes, though, would in effect cause a lower noise threshold and increase the likelihood of noise being mistaken for speech information.

5. CONCLUSION

5.1 SUMMARY OF RESULTS

The overall conclusion one reaches upon examining the results of the performance evaluation tests is that the voice input device, at the described level of development, is practical only with short, highly differentiated vocabularies or in special purpose applications (i.e., as a prosthesis device for the handicapped). For individual speakers the reliability of recognition of the digits (critical to any CAI system) approaches an acceptable level of 80 to 90% proper recognition. Nevertheless, with these rates a correction or conformation system would probably be necessary which might prove quite burdensome to the user. Children would have particular trouble with such a system since they tend to speak somewhat inconsistently and would be quite confused by errors in recognition.

The poor results of the multiple speaker tests indicate the necessity of specially training the system and keeping separate copies of vocabularies for each user of a lesson. This creates a problem in terms of memory space utilized. Even more importantly, large vocabularies would require that users spend a considerable amount of time with the training task, a requirement which may prove restrictive in many types of CAI applications.

A further limitation to the use of the speech input system is the processing time needed. With a 11.5 ms per vocabulary entry processing requirement, any situation requiring recognition of more than five or six words puts a strain on the system which is likely to result in automatic interrupts ("auto breaking"), causing a considerable delay before a response is sent.

One possible application of a voice input system with these limitations would be in conjunction with an audio output device to allow PLATO to act

as a workbook for students learning to service mechanical equipment. The audio output would be used to provide instruction to the user and the input would be used to "turn pages", request help sequences and issue other commands while the student is actually handling the equipment.

There are a number of conclusions to be drawn from the results of hardware modification of the original speech recognition system.

First is that a finish key with a more accurate duration measurement should be included in any system of this type. Being able to utilize an accurate duration measurement in the scoring of words resulted in noticeably improved recognition at a cost of only one additional key. The finish key provides the additional benefit of indicating the end of an utterance, allowing multiple utterances, and eliminating the need for silence at the end of a response.

The more accurate duration can be further utilized to smooth out timing errors in the reconstructed waveforms. By comparing the measured duration to the duration calculated from the key timing information, an error can be found which should be averaged into the 20 ms or greater timing separations, since they are the least accurate.

One can conclude from the results of both the two pass threshold system and the use of a slower clock that the system of sampling and key generation used originally does not result in as much spurious information as was speculated. These two systems, while substantially reducing the number of keys generated, also apparently caused the elimination of enough important information to cause more erratic recognition reliability.

There is another possible explanation for the erratic results stemming from these modifications, though. Since less keys are being generated, the spacing between keys becomes greater, hence the timing information becomes

less accurate, resulting in poorer recognition. It is possible that the increased accuracy afforded by the finish key measurement could negate the effect of this problem.

Exploration into the best set of individual threshold values to use leads to a number of possible sets of thresholds. The original set of 111 (using one for all measurements) seems to lie at the point where the change in reliability vs the cost in number of keys transmitted is at an optimal point, at least for the zero crossing measurements. Changes in threshold for the energy measurement had little effect on reliability yet had a noticeable effect on the number of keys generated. Setting the threshold to 3 could result in a 6 key per utterance saving (for the digits test) which could justify the cost of the additional circuitry necessary to allow different thresholds for different measurement types. Setting the bandpass filtered zero crossing measurement to 0 resulted in a substantial 13% increase in recognition over the rate with a threshold of 1, at a cost of 9 additional keys generated. This represents a considerable reliability improvement but it results in an additional delay in sending out keys (of .75 or 1.5 seconds depending on the device). As with the two pass modification the decreased number of keys stemming from high thresholds could result in less accurate timing information, so further testing should be performed with the finish key added.

The results of both the endpoint prediction circuitry and the logarithmic amplifier were inconclusive. The endpoint prediction circuit in particular did not behave as one might expect since there was no increase in key output rate or change in reliability. There are a number of explanations for the results obtained, including the use of a noise limiting microphone and the use of silicon diodes that cut out lower amplitude signals. I would recommend

further testing of these modifications with a different microphone having similar bandlimited frequency response to the Plantronics microphone (to avoid the passing of low frequency signals which mask the raw zero crossing measurements) but without the noise limiting circuitry. In addition, the silicon diodes should be replaced by germanium diodes or the voltage range used in the energy measurement should be increased.

5.2 SUGGESTIONS FOR FURTHER CHANGES AND RESEARCH

The energy measurement proved to make the least significant contribution to the recognition process. One reason for this is the limited range of the measurement compared to the wide range of speech intensities. By sending the difference between values measured and normalizing the result, one can extend the range of amplitudes. One drawback of this method is that since each measurement depends on the previous one, the loss of a key can cause serious errors. These errors can be corrected to a certain degree by keeping track of the overall error (by determining how much the final value differs from zero) and readjusting accordingly. An additional problem is that steep drops in intensity greater than the range of the energy difference measurement will cause errors in the absolute intensity from that point on.

Timing proved to be one of the nemesises of this system. The correlation between the calculated duration and the correctness of response was consistently low, except when the finish key was used. In many cases words were misrecognized because parts of the curves were displaced in time even though the overall shapes were the same as a vocabulary entry. Part of the timing problem seems to lie in the encoding of the timing information, which does not appear to be optimal. Using the number of sampling cycles occurring since the last key was sent could allow accurate measurements up to 30 ms.

By utilizing knowledge about the sampling sequence, on can further refine the accuracy of the measurement. For example, if the raw zero crossings are sampled right after the energy measurement, then if a raw zero crossing key is followed by an energy key with a 1 cycle indication in the timing bits, one knows that 13.3 ms have elapsed since the time for cycle is 10 ms, and the time between zero crossing and energy measurements are 1/3 that (3.33 ms). Lost keys, though, will reduce the accuracy and reliability of this method. Also, since samples represent events taking place over a 10 ms period, it is not clear that this sort of accuracy is necessary. A simpler system to implement would involve taking all three measurements at almost the same time and then waiting 10 ms to make the next burst of samples, rather than equally spacing the samples in time. This avoids logic that would be needed to keep track of what the last key type sent was.

The absolute gain of the voice input did not seem to make a significant difference in terms of recognition reliability over the limited range of gains tested, though the number of keys did decrease with decreasing gain. Nevertheless, one can speculate that variations in volume probably will have a significant impact on reliability. Therefore some sort of automatic gain control system might be desirable. A fast reacting gain control is not advantageous since such a system tends to shape the energy curve. Additionally, the AGC would have to be built such that background noise isn't amplified excessively when speech information is present. Any AGC system used therefore would have to be built to adapt to a speaker's volume level over a series of utterances and must lock at a particular gain whenever the speech signal is absent. One possible system was proposed by R. W. Scarr[7]. It involves having a 2 second time constant to allow for reacting to high energy sounds, a 9 second time constant for handling periods of low intensity and a 2 second

freeze time constant that determines when the system should be frozen after the intensity falls.

The raw zero crossing measurement of the current system does not seem to make as valuable a contribution to recognition as it might. Its value is diluted by the low resolution of the measurement resulting from its large range. In addition, the high correlation between the two zero crossing measurements indicates that redundant data is being sent, which runs contrary to the need to send only the most necessary information. A better approach may be to perform a high pass filtered measurement (from 3 to 6 kHz) in addition to the current bandpassed filtered measurement. This would still indicate whether fricatives are present and provide more accurate and unique information about the speech signal.

A considerable difference exists between certain characteristics of men's, women's, and children's speech. The frequency range of the second formant is particularly affected by the nature of the speaker. By adapting the passband of the filter used to isolate the second formant information, according to knowledge of the speaker, it is possible that the effectiveness of recognition will be increased. The band would be adjusted to the highest range for children and the lowest one for men. It is possible that a test word spoken by the user might allow the system to choose the proper band to use. To determine if this is the case, it would be beneficial to run a series of tests of male, female, and child speakers uttering words designed to isolate the effect of the zero crossing measurement (such as "spondee" words, like toothpaste and raincoat, that have two syllables of equal emphasis).

In the same way that small amplitude variations at low levels have more significance than larger changes at high levels, so small frequency variations

at low frequencies are more important than large ones at high frequencies. Hence, as with the energy measurement, it might prove advantageous to logarithmically compress the zero crossing measurements. This compression can be accomplished by increasing the resolution of the linear measurement and then performing a table lookup utilizing a read only memory to determine what value should actually be sent as part of the key. It is possible that a logarithmically compressed zero crossing measurement alone could provide enough significant information to allow effective recognition, since it enables one to have a large range with high resolution in the critical lower portion of the range.

A microprocessor (such as the 8080 in the new PLATO terminal) could be utilized to perform most of the hardware functions of the current speech input aside from the measuring and sampling tasks. It can make all decisions made by the comparison circuitry and use its random access memory to buffer the keys generated. In the case of the 8080 terminal, the sampling hardware can be attached to the I/O bus which can handle 25 K bytes/sec, well within the 2400 bits per second generated by the sampling system. The use of a microprocessor allows one to perform additional local processing of the speech signal. It can be used to monitor the amplitude and zero crossing rate of the ambient noise, in order to set noise threshold levels suited to the particular environment. In addition, the microprocessor can receive a 6 bit energy measurement and choose the 4 bit range that provides the best resolution (which in effect partially normalizes the measurement).

On a more powerful level a minicomputer could be used to perform all or part of the recognition processing locally, avoiding the need to send anything but a code indicating which response was chosen. Ideally a character string

representing the word would be sent which would make it appear to the system that the response was being typed, eliminating the need to modify lessons for voice input.

A problem may arise when it comes to down loading vocabularies. Since it currently takes 240 bits to represent a vocabulary entry, loading at a 960 bit per second rate allows one to load only 4 vocabulary words a second. However, at that speed one could load a 65 word vocabulary in the time it currently takes to load a PLATO character set.

Due to the large number of multiplies in the recognition software a hardware multiply would probably be necessary to insure reasonable processing speed. At 3 μsec per 16 multiply a mini could probably perform the processing in a shorter time than the central system.

Since a minicomputer would be attached directly to the speech input hardware, there is no limit to the amount of speech data that can be sent to the computer other than that imposed by the speed of the machine. Thus one could send a relatively large amount of high resolution data without having the bandwidth limitations presented by the PLATO communication system. Having such a capability would allow more accurate and wide ranging measurements than are currently made. Thus the availability of a minicomputer would probably allow a more sophisticated and better measurement system.

In conclusion, while some of the hardware modifications made to the voice input hardware device have resulted in performance improvements, the increased reliability was not sufficient to allow widespread application of this recognition system. Nevertheless, a number of areas of research exist that still may yield significant performance improvements and a number of minor hardware changes could affect the impact of some of the modifications, yielding a more workable system.

REFERENCES

1. S. Smith and B. Sherwood, Educational Uses of the PLATO Computer System, to be published.

2. J. Parry, Private Communication.

3. C. Neroth, Audio Graphic Programming System, Thesis, University of California at Berkeley (1972).

4. N. R. Ito and R. W. Donaldson, Zero Crossing Measurements for Analysis and Recognition of Speech Sounds, IEEE Transactions on Audio and Electro-Acoustics (Sept. 1971).

5. L. R. Rabiner and M. R. Sambur, An Algorithm for Determining the End Points of Isolated Utterances, Bell Laboratories (1975).

6. P. Denes and E. Pinson, The Speech Chain Anchors Book (1973).

7. R. Scarr, Normilization and Adaptation of Speech Data for Automatic Speech Recognition, Int J. Man-Machine Studies (1970).

8. B. Sherwood, J. Stifle, The PLATO IV Communications System, CERL Report X-44 Computer Based Education Research Laboratory.

APPENDIX 1

DETAILED DESCRIPTION OF SPEECH INPUT HARDWARE

Before the voice input is able to process any speech information, it must receive an external output word that clears the FIFO, determines gain and sets the difference threshold levels. The input receives this word serially from PLATO, performs a serial to parallel conversion and latches the result. The three gain bits each control an analog switch which is connected across one of three resistors in the feedback loop of an operational amplifier. These resistor values were chosen to allow seven different levels of gain.

Once the analog speech signal has been amplified, the three measurements are made concurrently. The energy measurement is made by rectifying the signal and integrating over a 10 ms period with a second order integrator. The resulting dc signal is then digitized with a tracking analog to digital converter. The tracking A/D converter consists of a comparitor that compares the integrator output with the output of a digital to analog converter that has its input connected to an up/down counter. The "up/down" input of the counter counts up or down to track variations in the integrator level. The counter output thus represents the digital inverse of the integrator voltage. When the measurement is sampled, the counter clock is gated momentarily to freeze the output value. The output bits of the converter are connected to a 4 input NAND gate so that whenever all the bits are one (representing the lowest bound of the A/D converter) an "inhibit" flag is set to zero.

The raw zero crossing measurement is made by amplifying the signal to the point of clipping, putting the result through a Schmitt trigger, dividing the resulting pulse train by four with a rate multiplier and then counting

the pulses with a counter. During sampling the counting process is frozen by gating the pulse train with a NAND gate, after which the counter is reset. If the inhibit signal has gone low during the previous sampling cycle a flip flop (which is clocked by the freeze pulse) is set, which inhibits the Schmitt trigger output.

The bandpass filtered zero crossing measurement is made in a similar manner except that the signal is filtered with a second order active filter in the range of 1 to 3 kHz and the pulse train is divided by two rather than by four.

In the comparison section the measurements made on the speech signal are sampled and compared to the previously transmitted value that is stored in a tri-state buffer. The outputs from the three measurement sections are fed into a multiplexer that is controlled by a two bit clock that also inputs the type bits to the FIFO. The latches are sequentially activated by the freeze pulses corresponding to the various measurements and are compared to the current output of the multiplexer. This comparison is made by taking the difference between the two values with a subtraction circuit composed of an adder and a set of exclusive or gates, and feeding the result into a 4 bit comparitor that has the difference threshold set by PLATO as its other input. The resulting "delta" signal is high whenever the threshold is exceeded.

The "delta" signal is then latched by a flip flop which has a clock (ϕ) that pulses during the freeze pulse. This flip flops output ("SI") is in turn used to set a load FIFO flag that causes a key to be gated into the FIFO if it is not full. The "SI" signal is also used to cause the new measurement value to be loaded into the appropriate tri-state latch.

Once the FIFO receives a key, a FIFO data ready flag is set which in turn is used to set another flip flop that provides a Data Ready signal for PLATO. This data ready flip flop is clocked by a key per second key clock. The Data Ready is reset low by the PLATO Data Accepted signal once the key has been properly received by the terminal.

APPENDIX 2

HARDWARE ADDITIONS FOR THE PERFORMANCE EVALUATION TESTS

When the voice input is being used under normal conditions, a key clock is used to send out keys at a regular rate by clocking a flip flop thats output is used as the data ready flag. To insure that keys are not lost due to software interrupts, a handshaking scheme was implemented in conjunction with the performance evaluation system. This scheme utilized a bit in the PLATO output word to replace the key clock. When the key input software had completed its task, two external words were sent cycling the 11th bit of the output word up and down to form a clock pulse, which set the data ready high if the FIFO contained data. Thus, new keys were sent only after the processing of the previous key had been completed.

To guarantee that words on the tape recorder weren't missed due to processor delays, the tape recorder was turned on and off by the recognition software. Another bit of the external output word was used to accomplish this task. This bit (which was stored in a latch) was used to control a 5 volt relay that switches the 120 volt line to the tape recorder. Optical isolation was used to prevent noise from the relay from effecting the voice input logic. Everytime a utterance was required, the tape recorder was turned on and a 5 second start up allowed, to enable transients to settle before the FIFO clear was reset.

APPENDIX 3

DETAILS OF HARDWARE MODIFICATIONS

The circuit for the two pass key generation scheme was inserted between the measurement comparison section and the FIFO control circuitry. A set of JK flip flops are used as memory elements to indicate whether the difference threshold was exceeded on the last sampling pass as indicated by the value of the "delta" signal during the clock pulse. The following table illustrates the possible sequences of events that might take place with this circuit.

delta	Q_t	Q_{t+1}	ϕ'	SI
1	0	1	0	0
1	1	0	⎍	1
0	0	0	0	0
0	1	0	⎍	0

The flip flops are each clocked by the ϕ clock NANDed with one of the freeze pulses so that each flip flop responds to a particular measurement type. The flip flop outputs are then used to gate the clock only when the previous value of the corresponding measurement exceeded the threshold. This ϕ' clock in turn sets the "SI" flip flop which causes a FIFO load pulse only if the current "delta" is high. As a result, the threshold has to be exceeded on two consecutive sampling passes to cause the FIFO to be loaded.

The duration measurement contained in the finish key is obtained by gating a 50 Hz clock with the noise inhibit signal coming from the energy measurement section. The resulting pulse train is counted with a digital counter yielding the duration that the speech signal stayed above the threshold. The inhibit signal is latched using the 50 Hz clock to prevent momentary transistions of the inhibit flag from causing erroneous pulses from being counted. The output from this counter is attached to the multiplexer that controls the input to the 4 bit comparitor and the FIFO

input. This allows easy inputing of the duration information into the FIFO and also causes a load FIFO pulse whenever the "type" clock is set to 11.

These load pulses are inhibited by a gate that has one input connected to the output of a three input NAND gate that is low only when the type bits are both high and the output from a counter is low. This counter is allowed to receive a 20 Hz clock once the inhibit flag has gone low after it has already gone high (as indicated by the setting of a flip flop). When the counter reaches 8 after .4 seconds, the output of the three input NAND gate goes high allowing load FIFO pulses to be sent when the type clock is at 11, causing finish keys to be inserted into the FIFO.

The two threshold signals used in the endpoint determination circuit are easily produced in the measurement sections. The raw zero crossing threshold is produced by latching the highest order bit of the raw zero crossing counter, during the freeze pulse. The amplitude threshold is produced with a comparitor having a .05 volt reference, connected to the output of the integrator in the energy measurement section. The circuitry inhibiting the zero crossing measurements before the noise threshold is exceeded, is disconnected to allow measurements to be made continuously.

The actual endpoint prediction circuitry consists of two sections one section determines whether speech information is present and the other section takes a signal from the first and either clears the FIFO or inhibits key generation.

In the circuit that determines if speech information is present, the two thresholds are used as the input to a NOR gate. When either threshold goes low the output of the gate goes low which causes the output signal of that section to be set high and removes a clear signal from a 4 bit counter.

The counter then counts a 100 Hz clock signal. When the fourth bit of the counter goes high (after 80 ms) a monostable is triggered which in turn causes the output of that section to be set low and the counter to be cleared.

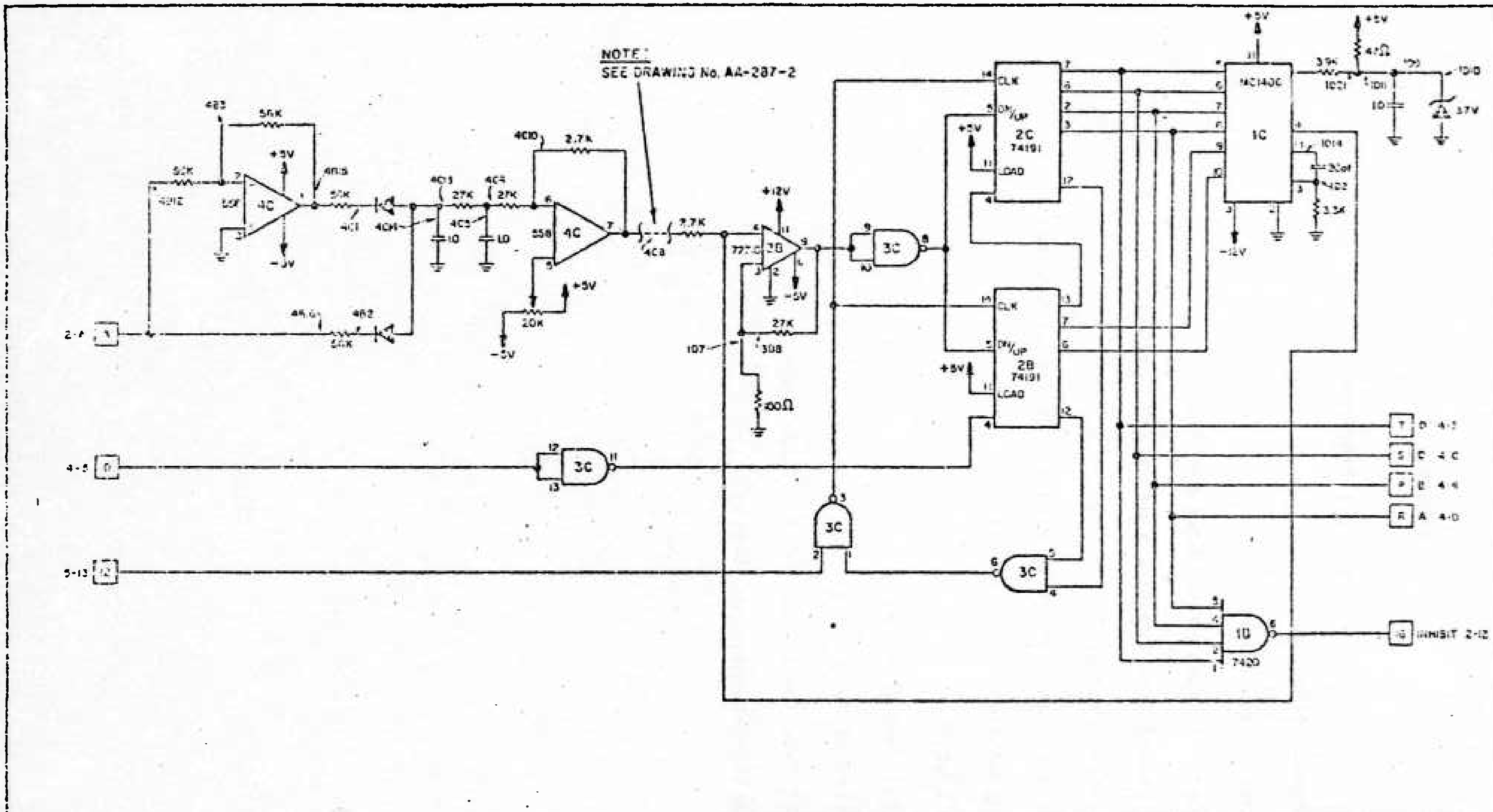
Before the inhibit goes high the Data Ready signal is inhibited with a NAND gate attached to a flip flop that is set high when the inhibit goes high. The inhibit signal is also attached to the clear of the counter. As a result, when the inhibit goes high, the Data Ready is allowed to go low enabling keys to be transmitted to PLATO. Also the inhibit going high prevents the counter from counting and triggering the monostable.

The signal from the first part of the circuit is then used for one of two purposes. Before the inhibit signal has gone high the signal is used to clear the FIFO and to clear the tri-state latches that hold the previous measurements. After the inhibit is gone high, the signal is used to set a flip flop which has its output connected to a gate that inhibits the load FIFO pulse, preventing further keys from entering the FIFO. Should the inhibit go high again this flip flop is cleared allowing more keys to be entered.

APPENDIX 4

CIRCUIT DIAGRAMS

List of Circuit Diagrams

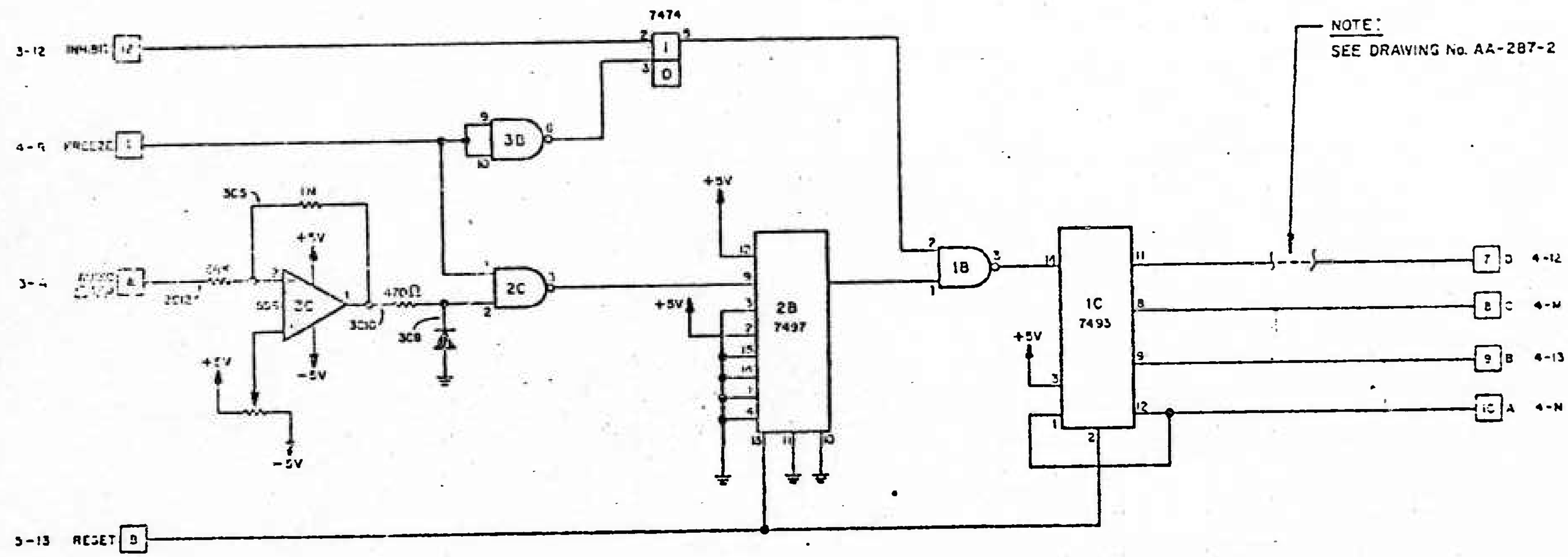


CARD No. 1 - VOICE INPUT DEVICE
INTENSITY MEASUREMENT SECTION

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS

TOLERANCES	DRAWN BY	SCALE
FRACTIONAL ±	RTL	
2 PLACE DECIMALS ±	APPROVED BY	DATE
3 PLACE DECIMALS ±		11-11-75

AA-277-2



CARD No. 2 – VOICE INPUT DEVICE – RAW ZERO CROSSING MEASUREMENT SECTION

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS

TOLERANCES
FRACTIONAL ±
3 PLACE DECIMALS ±
2 PLACE DECIMALS ±

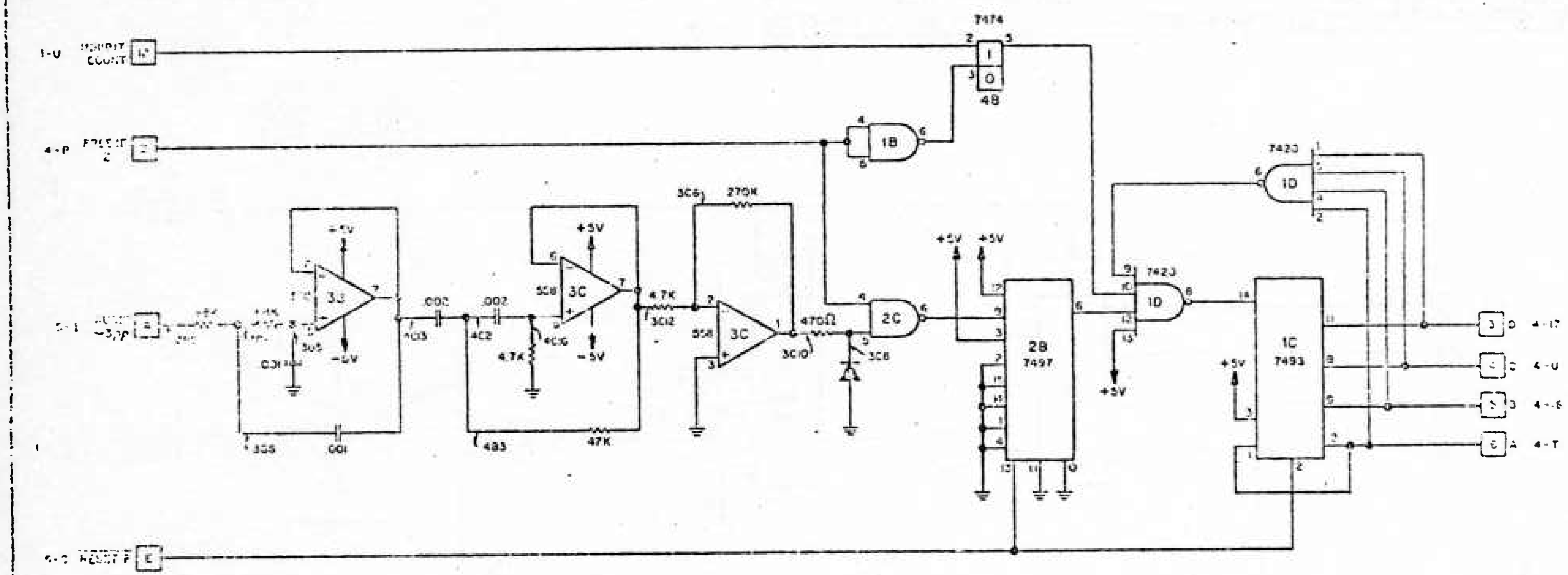
DRAWN BY RTL

APPROVED BY

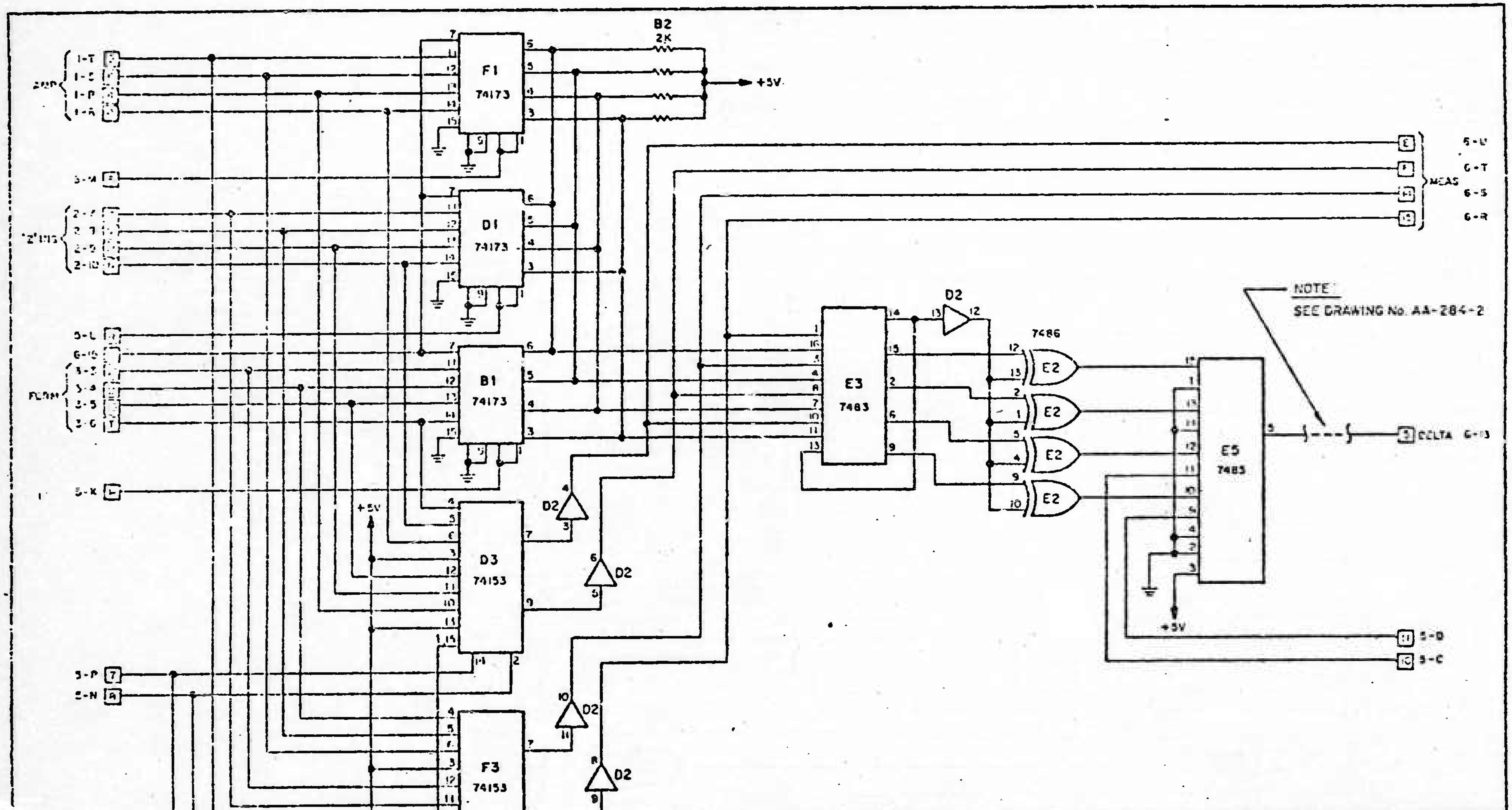
SCALE ———

DATE 11-12-75

AA-278-2



CARD No. 3 – VOICE INPUT UNIT BANDPASSED ZERO CROSSING MEASUREMENT SECTION			COMPUTER-BASED EDUCATION RESEARCH LABORATORY UNIVERSITY OF ILLINOIS
TOLERANCES FRACTIONAL ± 3 PLACE DECIMALS ± 2 PLACE DECIMALS ±	DRAWN BY RTL APPROVED BY	SCALE DATE 11-12-75	AA-279-2

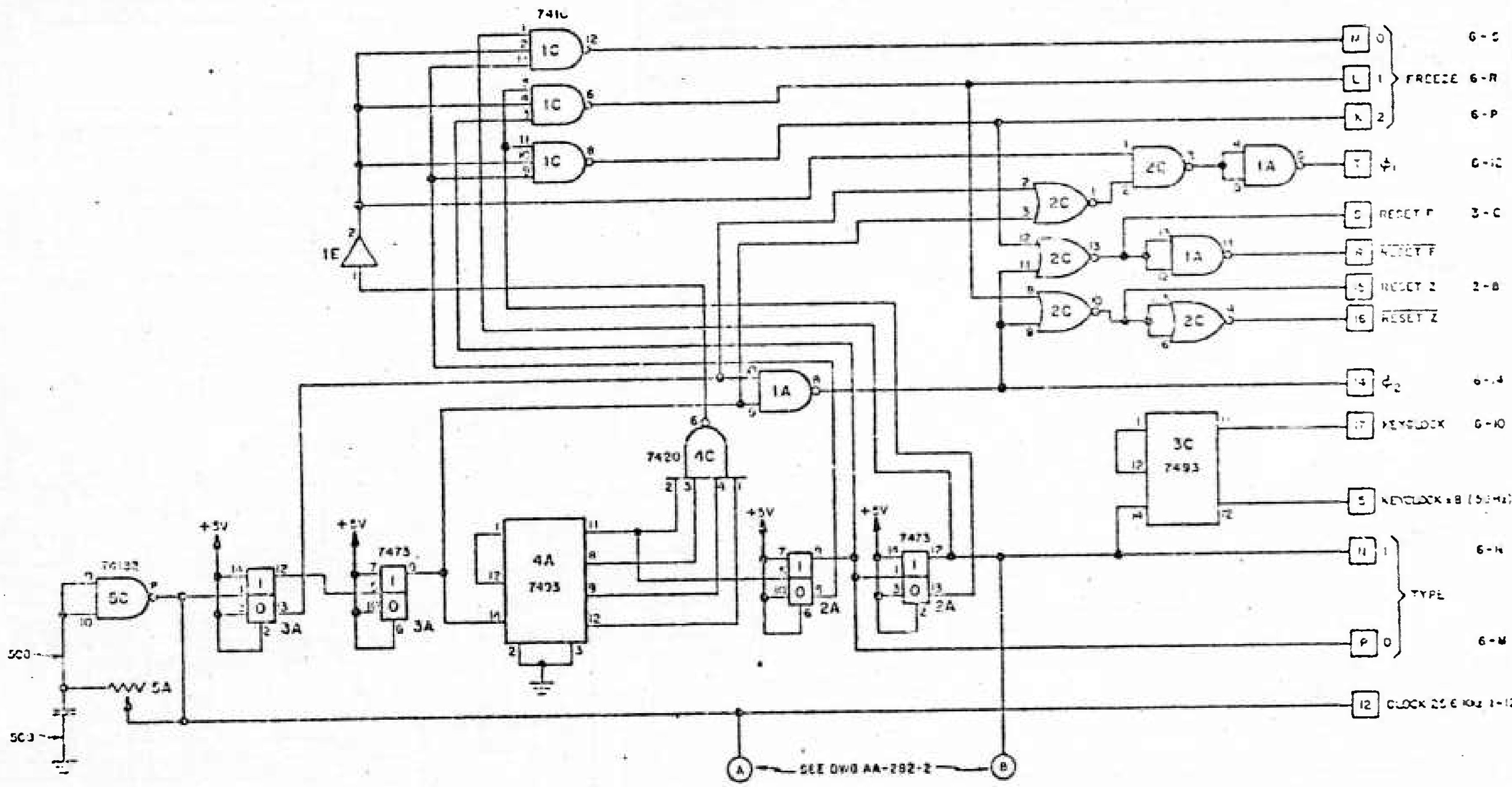


CARD No. 4 – VOICE INPUT DEVICE
COMPARISON SECTION

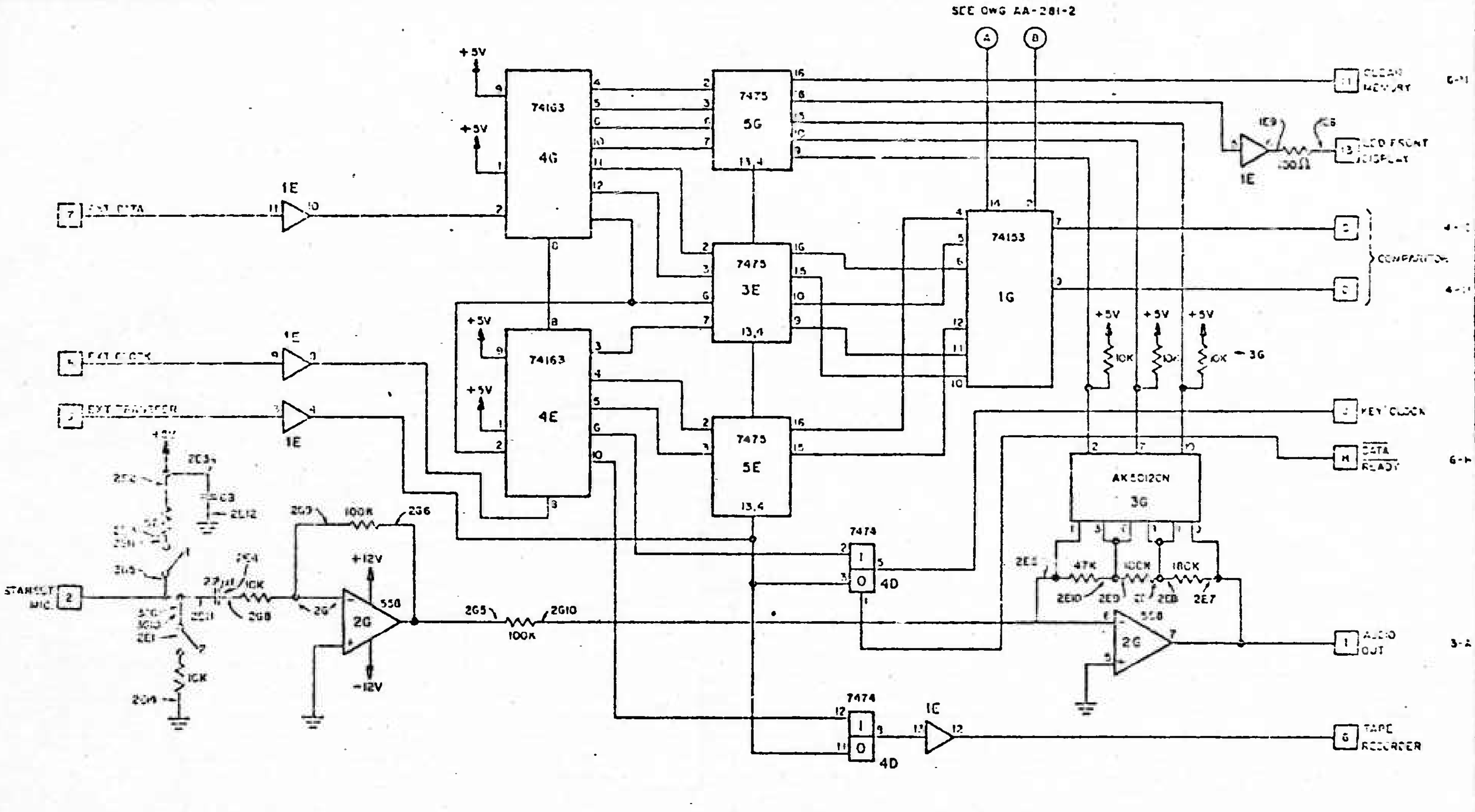
TOLERANCES	DRAWN BY RTL	SCALE
FRACTIONAL ± 3 PLACE DECIMALS ± 2 PLACE DECIMALS ±	APPROVED BY	DATE 11-13-75

COMPUTER-BASED EDUCATION
RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS

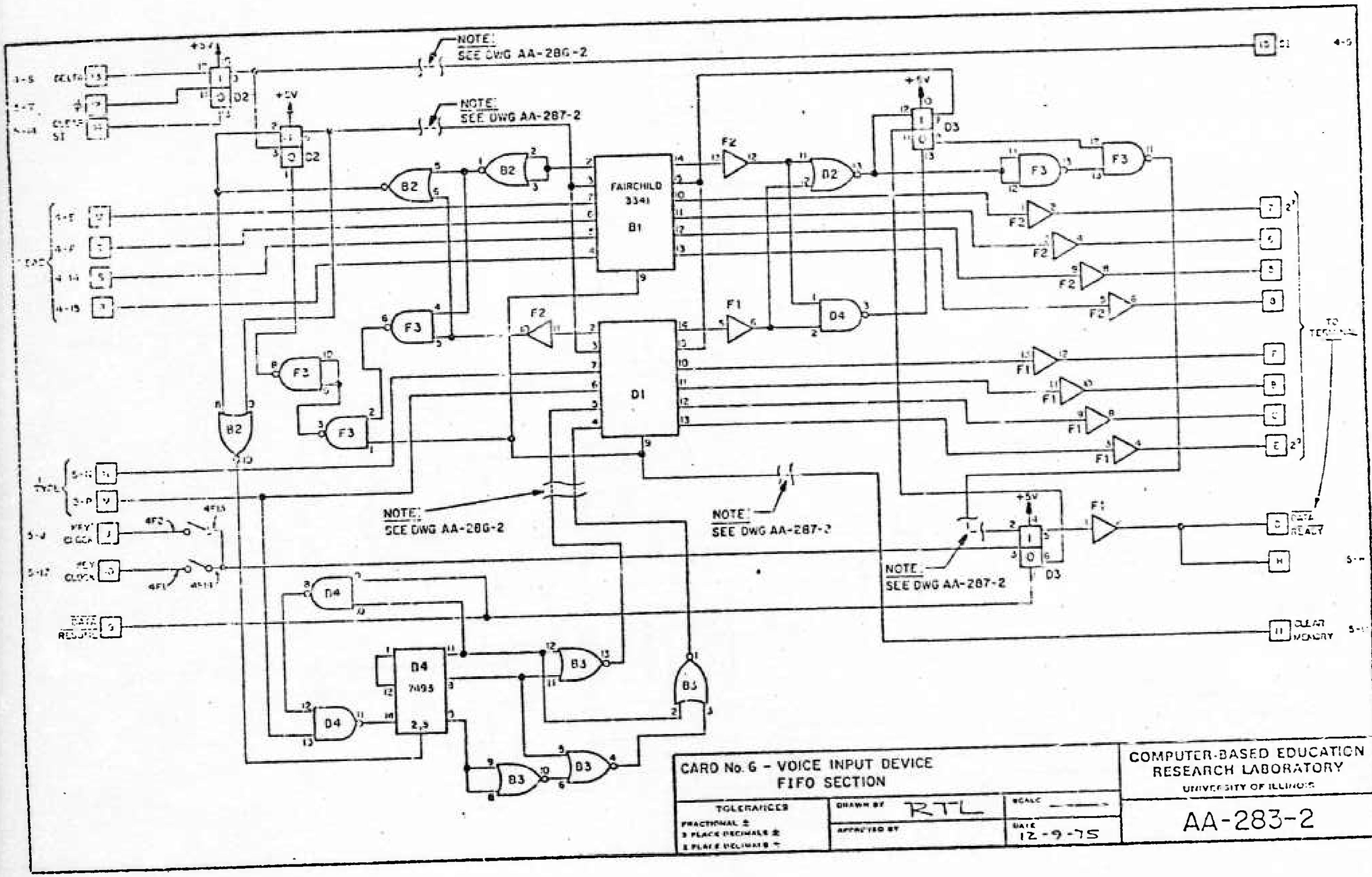
AA-280-2

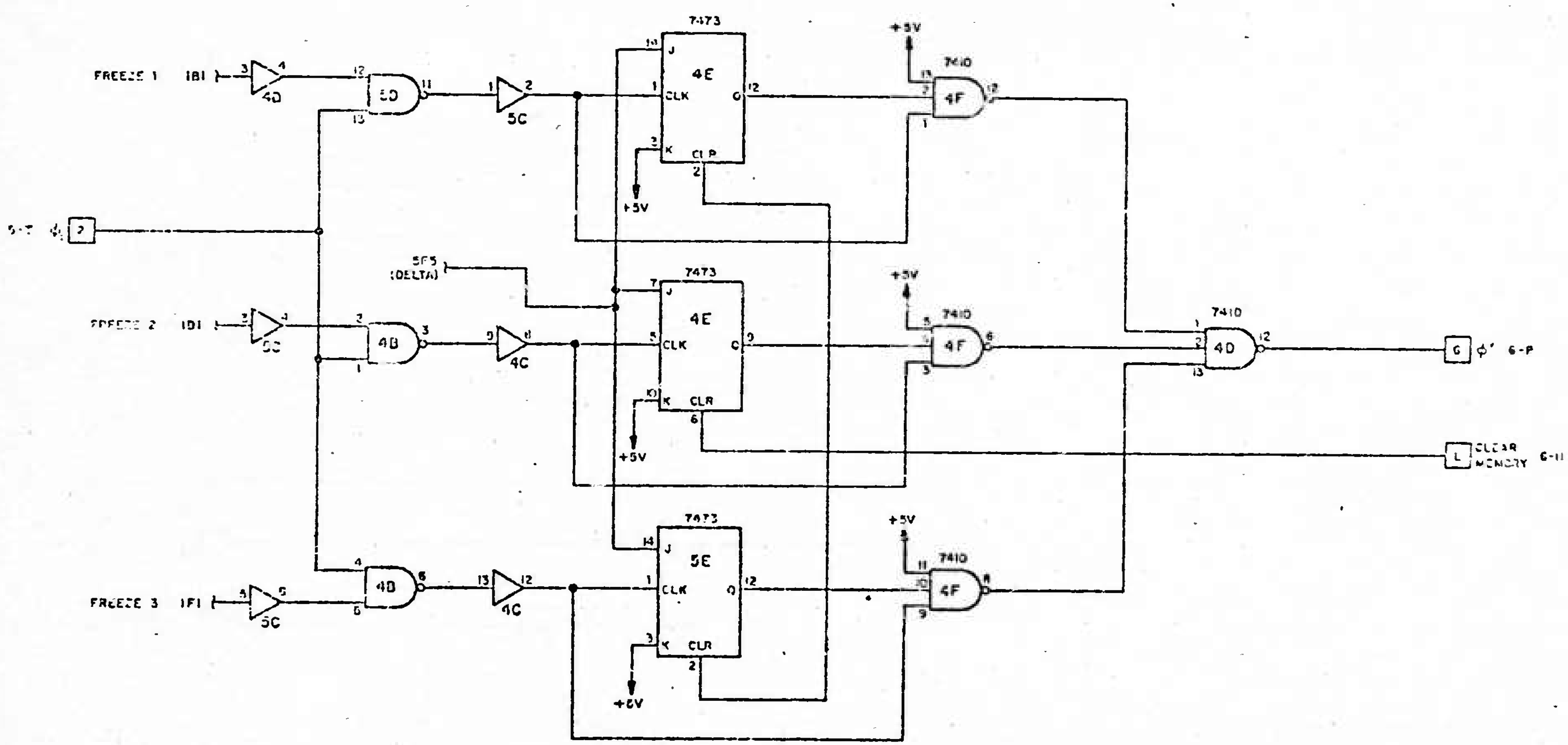


CARD No. 5 - VOICE INPUT DEVICE TIMING SECTION			COMPUTER-BASED EDUCATION RESEARCH LABORATORY UNIVERSITY OF ILLINOIS
TOLERANCES FRACTIONAL ± 2 PLACE DECIMALS ± 3 PLACE DECIMALS ±	DRAWN BY RTL	SCALE	AA-281-2
	APPROVED BY	DATE 12-8-75	

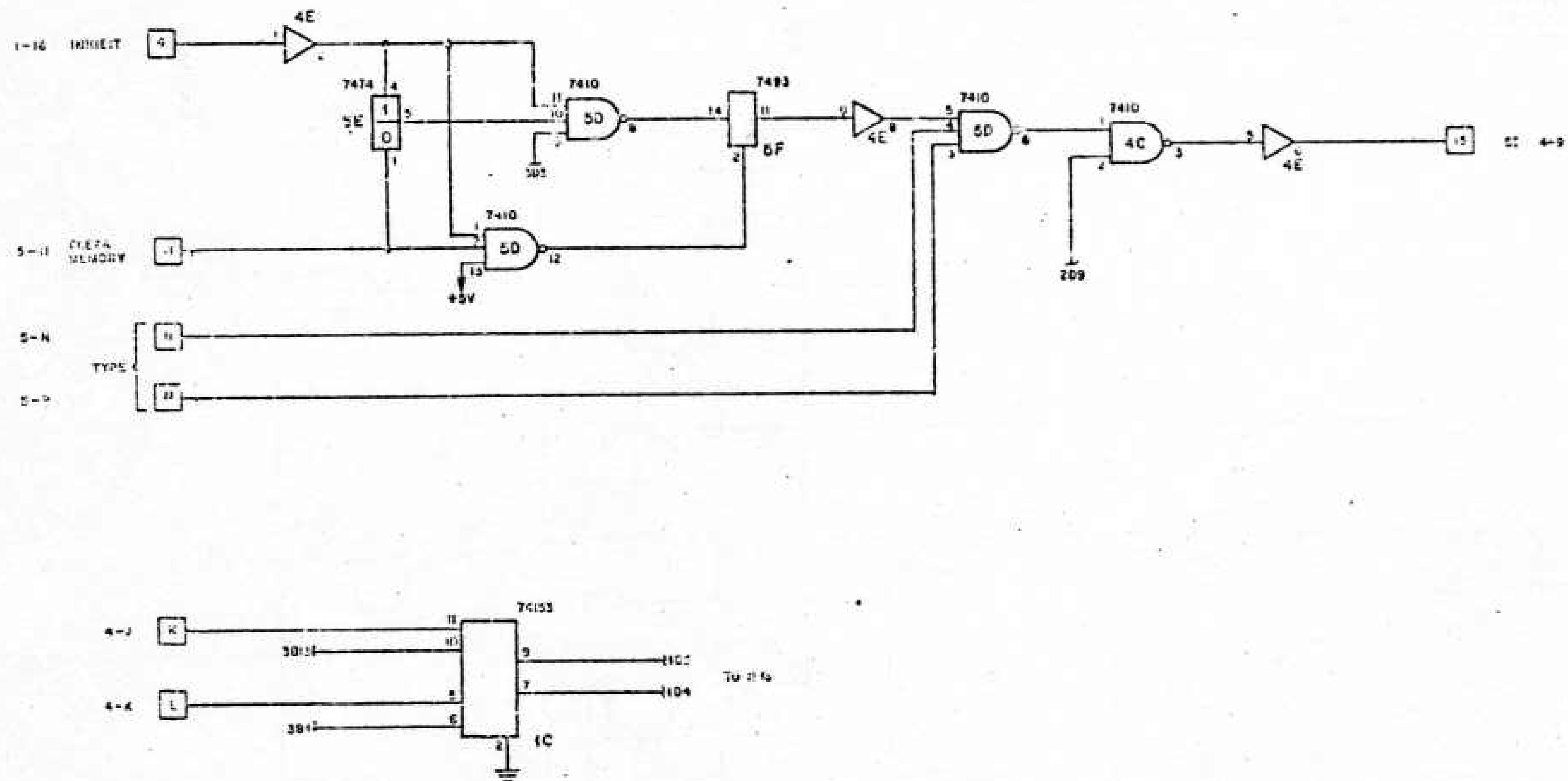


CARD No. 5 — AUDIO INPUT AND SERIEL TO PARALLEL CONVERTER			COMPUTER-BASED EDUCATION RESEARCH LABORATORY UNIVERSITY OF ILLINOIS
TOLERANCES FRACTIONAL ± 3 PLACE DECIMALS ± 2 PLACE DECIMALS ±	DRAWN BY RTL	SCALE ———	AA-282-2
	APPROVED BY	DATE 12-8-75	


NOTE: SEE DWG AA-286-2
NOTE: SEE DWG AA-287-2
FAIRCHILD 3341 B1
D1
D4 7493
NOTE: SEE DWG AA-286-2
NOTE: SEE DWG AA-287-2
NOTE: SEE DWG AA-287-2
DELTA
KEY CLOCK
DATA REQUEST
DATA READY
CLEAR MEMORY
TO TERMINAL
CARD No. 6 - VOICE INPUT DEVICE
FIFO SECTION
TOLERANCES
DRAWN BY RTL
DATE 12-9-75
COMPUTER-BASED EDUCATION
RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS
AA-283-2



ADDED TO CARD No. 4 — VOICE INPUT DEVICE TWO PASS MODIFICATION			COMPUTER-BASED EDUCATION RESEARCH LABORATORY UNIVERSITY OF ILLINOIS
TOLERANCES FRACTIONAL ± 3 PLACE DECIMALS ± 2 PLACE DECIMALS ±	DRAWN BY RTL	SCALE	AA-284-2
	APPROVED BY	DATE 11-13-75	



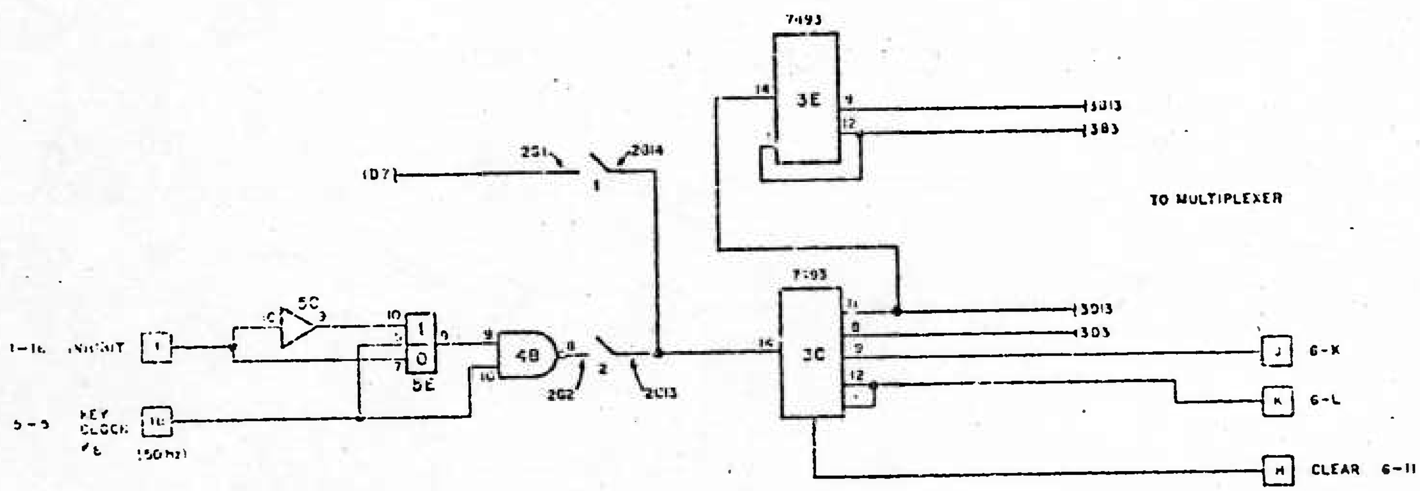
CARD No.G – VOICE INPUT, END OF DURATION INDICATION FOR FINISH KEY

COMPUTER-BASED EDUCATION RESEARCH LABORATORY
UNIVERSITY OF ILLINOIS

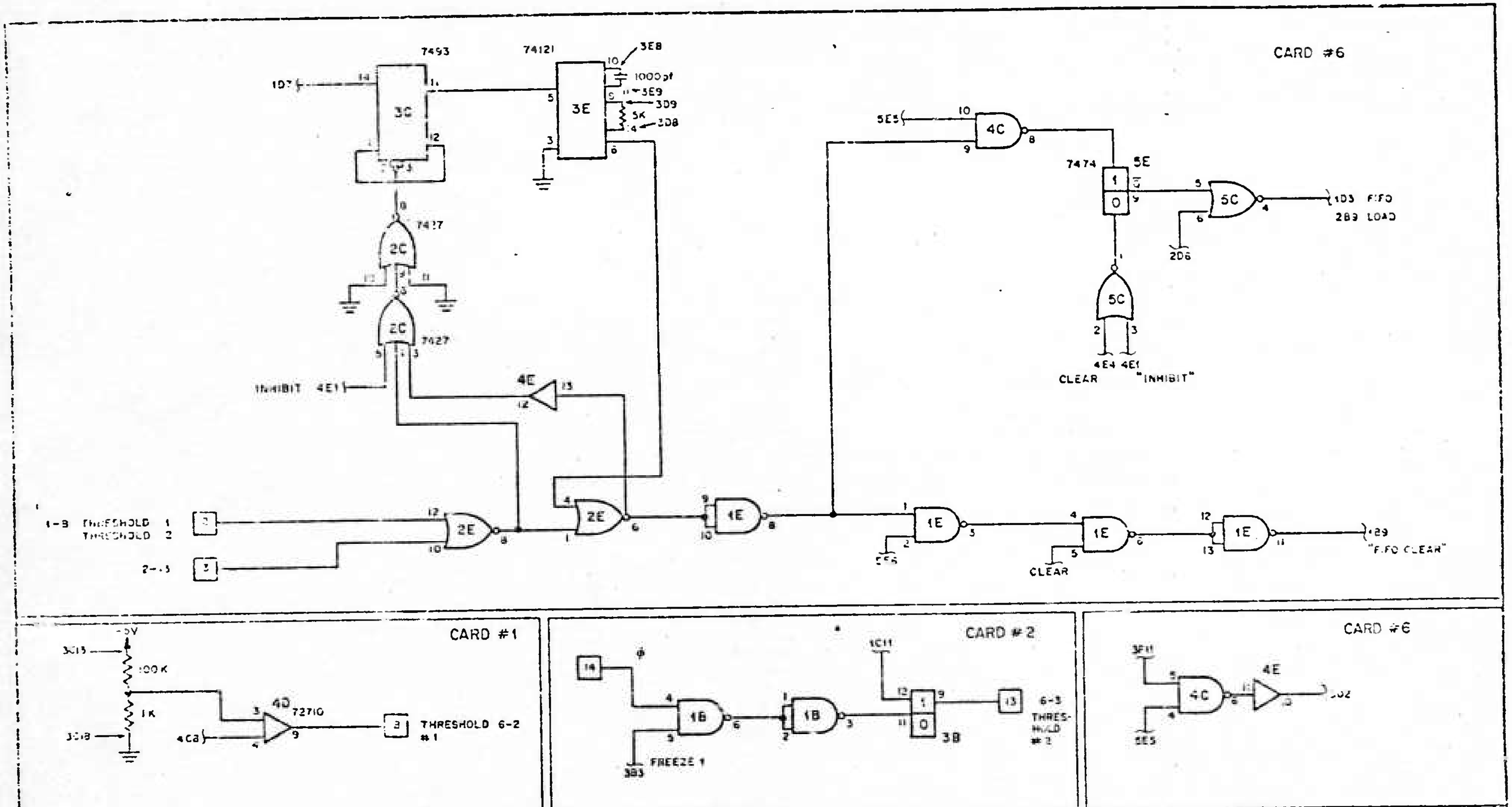
TOLERANCES
FRACTIONAL ±
2 PLACE DECIMALS ±
3 PLACE DECIMALS ±

DRAWN BY W. W.
APPROVED BY
SCALE
DATE 12-9-75

AA-286-2



CARD No. 4 — VOICE INPUT DURATION OR KEY COUNTER			COMPUTER-BASED EDUCATION RESEARCH LABORATORY UNIVERSITY OF ILLINOIS
TOLERANCES FRACTIONAL ± 3 PLACE DECIMALS ± 4 PLACE DECIMALS ±	DRAWN BY W.W.	SCALE	AA-285-2
	APPROVED BY	DATE 12-3-75	



VOICE INPUT
ENDPOINT DETERMINATION SECTIONS

TOLERANCES FRACTIONAL ± 3 PLACE DECIMALS ± 2 PLACE DECIMALS ±	DRAWN BY W. W. APPROVED BY	SCALE DATE 12-9-75	COMPUTER-BASED EDUCATION RESEARCH LABORATORY UNIVERSITY OF ILLINOIS AA-287-2